# Interpreting Emotion from Sound:
# A Comparison of Features and Models for Music Emotion Classification

By
Shaun Leib

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented May 16, 2025
Commencement June 2025

# AN ABSTRACT OF THE THESIS OF

Shaun Leib for the degree of Honors Baccalaureate of Science in Computer Science presented on May 16, 2025. Title: Interpreting Emotion from Sound: A Comparison of Features and Models for Music Emotion Classification

Abstract approved:

---

Patrick Donnelly

The classification of emotion in music is a task of recent importance in the field of music information retrieval. However, the current state of the art has stagnated with very few breakthroughs. In our work, we analyze a large set of extracted acoustical features in an attempt to find a subset of emotional relevance. Utilizing the 4Q dataset with ground truth values corresponding to the four quadrants of Russell's valence and arousal model for emotion, we obtain these features using five different extractor toolkits. We employ various feature selection methods to find the most important feature types by comparing the results of different machine learning models. We compare our results to the best results on this dataset to date. We find that our feature sets contain more emotionally relevant information, and our models are able to learn more complex relationships when predicting an emotion.

Key Words: Music Information Retrieval, Music Emotion Recognition, Feature Extraction, Deep Learning

Corresponding e-mail address: leibs@oregonstate.edu

©Copyright by Shaun Leib
May 16, 2025

# Interpreting Emotion from Sound:
# A Comparison of Features and Models for Music Emotion Classification

By
Shaun Leib

A THESIS

submitted to

Oregon State University

Honors College

in partial fulfillment of
the requirements for the
degree of

Honors Baccalaureate of Science in Computer Science
(Honors Scholar)

Presented May 16, 2025
Commencement June 2025

Honors Baccalaureate of Science in Computer Science project of Shaun Leib presented on May 16, 2025.

APPROVED:

---

Patrick Donnelly, Mentor, representing Electrical Engineering and Computer Science

---

Stephen Ramsey, Committee Member, representing Electrical Engineering and Computer Science

---

Jinsub Kim, Committee Member, representing Electrical Engineering and Computer Science

---

Toni Doolen, Dean, Oregon State University Honors College

I understand that my project will become part of the permanent collection of Oregon State University Honors College. My signature below authorizes release of my project to any reader upon request.

---

Shaun Leib, Author

# Contents


**1 Introduction** **2**
- 1.1 Automatic Emotion Classification . . . 2
- 1.2 Our Work . . . 3

**2 Background** **5**
- 2.1 Machine Learning Models . . . 5
- 2.2 Models of Emotion . . . 10
- 2.3 Feature Selection Algorithms . . . 12

**3 Related Work** **16**
- 3.1 Evolution of Models . . . 16
- 3.2 Feature Sets . . . 17

**4 4Q Dataset** **20**
- 4.1 4Q . . . 20
- 4.2 State of the Art . . . 21

**5 Feature Set Extraction** **22**
- 5.1 Melody . . . 23
- 5.2 Harmony . . . 26
- 5.3 Rhythm . . . 28
- 5.4 Dynamics . . . 30
- 5.5 Tone Color . . . 33
- 5.6 Expressivity . . . 39
- 5.7 Texture . . . 41
- 5.8 Form . . . 41
- 5.9 High level and Voice . . . 42

**6 Experiments** **43**
- 6.1 Replication Study . . . 43
- 6.2 Other Inputs . . . 44
- 6.3 Feature Selection . . . 45
- 6.4 Models Based on Features . . . 47

**7 Discussion** **50**
- 7.1 Limitations . . . 51
- 7.2 Future Work . . . 52
- 7.3 Conclusion . . . 53

**8 Appendix** **54**

# 1 Introduction

Music Emotion Recognition, or MER, applies computational methods to identify and interpret the emotional content elicited in the listener of a song. It combines the knowledge from multiple fields including music theory, psychology, signal processing, and artificial intelligence. Automatic mood detection has many advantages over manual annotation. Historically, this type of annotation has been through careful human subject studies in a research lab or through crowd-sourced efforts like Amazon's Mechanical Turk, however, these methods are time-consuming and costly, and do not produce congruent results between independent annotators. This can be seen in datasets like DEAM [2], and AMG1608 [13], where the total amount of songs in these datasets is limited due to time and money, and there is a considerable amount of disagreement between annotators. These datasets are by no means flawed, in fact, they offer a lot of insight, however, they highlight the need for alternative automatic methods for emotion tagging for music.

## 1.1 Automatic Emotion Classification

Automatic emotion classification has broad applications across many different spaces. These include music recommendation systems, mood based playlist generation [18], affective music therapy [27], and generative music composition [16]. MER plays a vital role within the broader domain of music information retrieval, or MIR, aiming to bridge subjective human perception and machine understanding of music.

With these considerations in mind, there have been many efforts to design such methods. These previous attempts have applied various approaches to the problem. Firstly, the input type is of great consideration. Since raw audio waveforms are extremely complex and high-dimensional, it is difficult for computational models to distinguish relevant emotional patterns from noise. Therefore, other input types such as acoustical features [49] [42] [71] [14] [69], lyrics [70], spectrograms [54] [14], as well as combinations of these [18] have been utilized. Additionally, many different types of machine learning models have been utilized to process these inputs ranging from older conventional models such as SVMs [49] to more modern deep learning approaches [40] [19] [39]. Finally, various emotional taxonomies have been applied, ranging from arbitrary classes to psychologically backed models.

However, as pointed out by Scholz, Ramalho, and Giordano, progress in many MIR fields have stagnated. They argue that this is due to approaches lacking in musically knowledge intensive techniques, and the lack of deep learning [57]. Panda et al. acknowledge this in [49] and further argue that the difficulty in MER stems from ambiguity in human perception of emotion, the absence of a public widely accepted dataset with a psychologically backed taxonomy, and most importantly, the lack of relevant acoustical features designed for emotion classification.

To address these issues, Panda et al. created a new dataset utilizing four quadrants

of the valence and arousal graph. With this dataset, they extracted acoustic features from each song, and performed ReliefF feature selection to find the top 100 features for emotion classification. Since the dataset only contains four classes, the task of predicting these emotions should be simple, however, the state of the art for this dataset is only able to obtain a 76.4% $F_1$ score [49].

## 1.2 Our Work

In our work, we aim to break the stagnation in MER. We present a comparison of machine learning models which we utilize to predict emotion from a large corpus of acoustic features. Our study builds on the work done by Panda et al. by incorporating a large amount of novel features not present in their study which we show to be emotionally relevant. This is accomplished by utilizing more feature extraction toolkits and testing different feature selection methods to determine the features which have the highest contribution to classification. Finally we apply stronger, more expressive deep learning models in order to capture this relevant information.

In our experiments, we restrict the majority of our features to those which describe the audio, and not the lyrics or vocals. While lyrics and vocals do affect the emotion of a song, this analysis is beyond the scope of this project. This is done because in the field of MER, acoustical features and lyrical features are two separate domains which are simultaneously being researched.

We start with background information in Section 2 with a detailed view of many of the systems in this study. These include an overview of the different types of machine learning models, psychological models of emotion, and feature selection methods. We continue to provide background information in Section 3, where we give analysis the on past studies in music emotion recognition that have led to our paper. We split this section into different machine learning models that have been used and a background in acoustical feature sets. Next, in Section 4, we describe the dataset employed in this study. In Section 5, we discuss both the process which we utilized to extract acoustical features from the audio and the relationship between emotion and these features. We detail eight musical dimensions (Melody, Harmony, Rhythm, Dynamics, Tone Color, Expressivity, Form, High Level) which we use to classify our features.

In the experiments section, Section 6, we begin by replicating the study Panda et al. completed, and we achieved comparable results. Next, we demonstrate the difficulty of the problem by utilizing raw audio waveforms and audio spectrograms as input for neural networks. Our solution, in Section 6.3, employs feature selection methods on the feature set we extract to obtain the most emotionally relevant subsets. In the subsequent section, we evaluate these subsets on support vector machines for a baseline, and for comparison against Panda et al. We then use other machine learning models and varying feature subset sizes to obtain our best results, outperforming the state-of-the-art on the dataset. Finally, in Section 7, we discuss our contributions, limitations, and necessary future work.

To our knowledge, our best performance, 82.03% $F_1$ Score, is the best result so far on this dataset. We accomplished this with the use of a multilayer perceptron trained on 300 total features. In order to validate these results, we replicate the experiments performed by Panda et al. by utilizing their 100 features. In all cases, regardless of model or size, our feature set outperforms theirs.

# 2 Background

In this section, we provide preliminary information necessary to complete our work. We describe the different types of machine learning models used, the different models of emotion, and the different feature selection algorithms we employ.

## 2.1 Machine Learning Models

In our research, we compared multiple different types of machine learning model architectures. The models we explored have previously demonstrated the ability to represent the emotional space of music [49] [39] [54]. Additionally, we chose models to both recreate results from previous experiments, and newer models against which we compare them. These models are all supervised machine learning models, meaning that we have a previously labeled set of training data we use to train our model before testing. This allows our model to learn the relationship between our music input and our emotional output. Where the models differ is in the need to process different formats of information as input and how these inputs are processed, and the details will be explained in each respective section below.

### 2.1.1 Support Vector Machines

The first model we employ is a Support Vector Machine, or SVM. This is an older type of machine learning model, and operates by generating decision boundaries in a high dimensional space to determine outputs. These decision boundaries are determined by maximizing the margin distance between two classes. This is done by identifying the support vectors, or the two closest instances of the two classes. The original models, created by Vapnik and Cortes, could only generate linear boundaries and did not account for outliers. Later versions incorporated kernel functions which transformed the inputs into higher dimensional spaces allowing for nonlinear decision boundaries, and the addition of slack variables allowed for "soft-margins" and some misclassifications [15]. In later work, researchers adapted the support vector machine concepts and applied them to regression tasks. This was accomplished by approximating a function whose error, $\varepsilon$ , was bound by the training data instead of generating a hyperplane decision boundary which separated classes [1].

### 2.1.2 Random Forest

Random forests are an ensemble model which combines multiple decision trees together to make predictions. A singular decision tree is a type of weak learner which use tree structures for both classification and regression. The inputted data is fed to the root of the tree, and a series of conditionals recursively subdivide the input space until it reaches a leaf node which contains the resulting predictions. The deeper the tree, indicating more conditionals, the more complex functions can be represented by

it, but this can quickly lead to overfitting. For example, if the number of leaf nodes equal the number of training examples, then the model has simply memorized the training data. This trade off between complexity and memorization must be optimized, and the series of conditionals must be learned through training to determine the best configuration.

A decision tree is a weak learner, which means that the model will have low variance but high bias and cannot completely represent complex functions without overfitting. To combat this, an implementation of a random forest employs an ensemble method known as bagging. With bagging, many separate shallow decision trees are trained using a random subset of the dataset and a random subset of features at each decision. Finally, majority voting or averaging is used on the predictions of all the trees to obtain a singular label in the end. By using bagging, random forests are able to generalize more than decision trees [30].

### 2.1.3 Neural Networks (MLP)

The most basic model in deep learning is the multilayer perception (MLP). Capable of classification and regression, it is a feedforward network of fully connected neurons with non-linear activation functions which transform input data into different dimensions. A standard MLP contains an input layer, hidden layers, and an output layer. The input layer has the same dimensionality as the input, i.e. the same amount of neurons as features. Similarly, the output layer's dimensionality is determined by the amount of expected outputs. For example, a classification problem with four classes would have four neurons, whereas a regression problem with a single output would contain one. Finally, the hidden layers are located in between the input and output layers. They are responsible for processing the information from the input layers and transforming the data until it reaches the output layer. A single neuron in a hidden layer gets its input from every neuron in the previous layer and sends its output onward. The choice of the amount of hidden layers, as well as how many neurons in each hidden layer is specific to each problem. A visual representation of this can be seen in Figure 1.

Training an MLP requires three steps, forward propagation, evaluation, and backwards propagation. In the forward step, inputs are sent through each layer in the model until it reaches the output layer. A weighted sum defined as:

$$a = \sigma(b + Wx)$$

is computed in order to determine what information gets sent on to the next layer. These weight $W$ and bias $b$ terms act as the connection strength between a neuron in one layer and all the neurons in the next, and they are learned, or adjusted, during the backpropagation step. Additionally, an activation function $\sigma$ is used between layers to transform the data into complex nonlinear spaces. Common activation functions include ReLU (Rectified Linear Unit), tanh, sigmoid, and softmax.

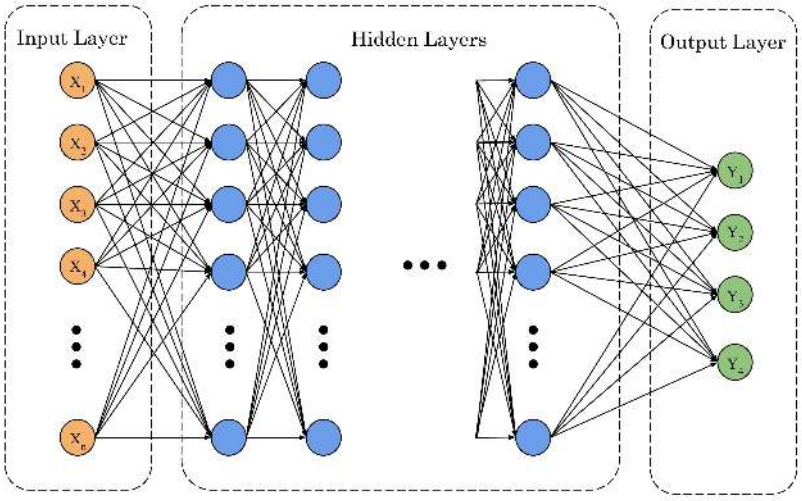



Figure 1: A visualization of the layers of a fully connected MLP. The network has $n$ inputs, and four outputs.

Now that the forward pass has completed, there needs to be a way to evaluate the output computed. Loss functions are used for this purpose as they allow the model to quantifiably measure how "bad" its output was. There are many different kinds of loss functions which are dependent on the type of problem. For classification problems common loss functions include binary cross-entropy and categorical cross-entropy. For regression problems, common loss functions include mean squared error and mean absolute error.

The ultimate goal of most machine learning algorithms is to find the best set of weights and biases to minimize the loss function computed in the previous step. A small value means the model accurately predicted or modeled the input information. This is where the final step, backpropagation is utilized. This step computes the gradient of the loss function with respect to the model's parameters using the chain rule. This calculation is propagated backwards through the network, and adjusts the model's weights and biases in the opposite direction, or steepest descent, of the gradient. The amount each is adjusted is controlled by the learning rate, a hyperparameter, set before running.

These three steps are repeated for every input in the dataset as well as for a predefined amount of epochs in order to continually minimize the loss function. By incrementally adjusting the model parameters, the network is able to learn complex patterns within the dataset, and eventually make accurate predictions on new data not in the training set.

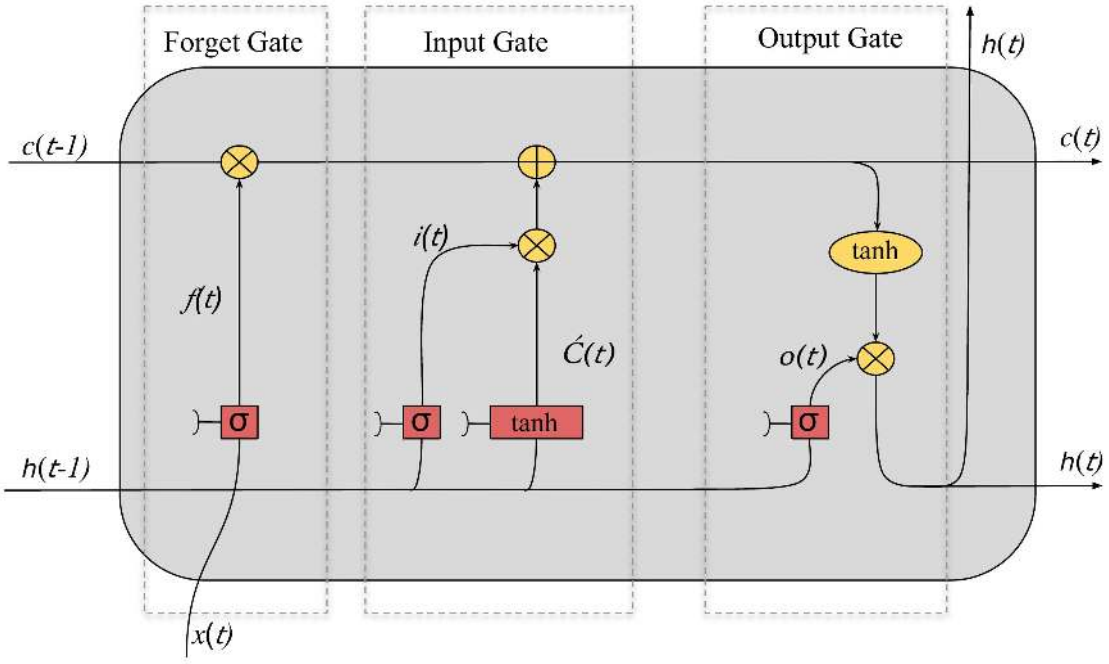



Figure 2: A visualization of an LSTM architecture, recreated from [72].

### 2.1.4 Recurrent Neural Networks

Recurrent Neural Networks are similar to MLPs however the neurons in an RNN layer are not independent of one another. Instead nodes contain self referencing cycles and share parameters across the layers of the network. These loops make RNNs specialized for sequential or time series data where each input to the network is a step. This is due to outputs being fed back and combined with the input of the next step, thus allowing previous intervals to influence the current one in the network. The mathematical expression [72] is similar to that of the MLP, but includes the previous output:

$$h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$$

where $x_t$, $h_t$ are the current input and outputs at time $t$. $W_h$, $W_x$, and $b$ are all trainable parameters. There are three different types of RNNs categorized by the type of input they receive and output they compute. *One to many* RNNs take a single input instance and produce many different outputs. For example image captioning taking a single image and produces a list of words. *Many to one* does the opposite, taking many inputs and computing a single output, often used in categorization or time series predictions. Finally, *many to many* takes multiple inputs and produces many outputs. Additionally, LSTMs use a slightly altered version of backpropagation called backpropagation-through-time to account for the sequential data [72].

However, RNNs are limited by both vanishing and exploding gradients, causing

larger models and large input lengths to under perform when the gradient becomes too small or large. These problems are addressed in LSTMs (Long Short Term Memory), a variation of RNNs. LSTMs add internal states to RNN nodes which also influence their decisions. This state, or cell, is able to remember contextual information for larger and more complicated sequential inputs. The cell contains three parts: forget gate, input gate, output gate as seen in Figure 2. Just like its name, the forget gate determines what information is kept in the cell state and deemed contextually relevant. The input gate decides what new information should be added to the state, and the output gate determines what parts of the current state should be considered to output. These gates have values [0-1] which determine how much or how little to let through (0 being no information and 1 being all). Because of this, LSTMs greatly increase the RNN's ability to "remember" information further in the past and help solve the gradient problems [72].

Finally, RNNs and LSTMs can be bidirectional. In this variant, a single layer has two independent states for both the forward and backward direction of the time series data, and both can be trained simultaneously. These two states operate the same as a regular RNN would, but the backwards state analyzes the data in reverse. This allows the model to understand the input information with both past and future context and generate better results [58].

### 2.1.5 Convolutional Neural Networks

The final type of artificial neural network discussed here is the convolutional neural network (CNN). Popularized by tasks in computer vision, these networks get their name from the mathematical operation of the same name. In a convolution, a learnable kernel function extracts features from a given input. These kernels are smaller in size and operate like a sliding window stepping across the input. While moving across the input, the dot product of the kernel weights and inputs are calculated and saved as seen in Figure 3.

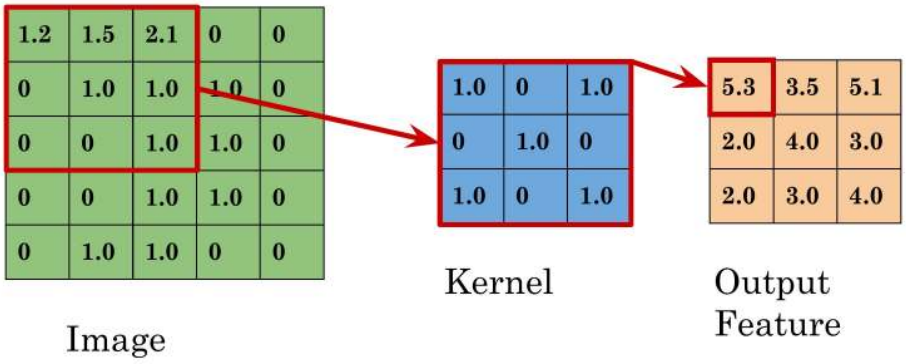



Figure 3: A visualization of a single convolution, recreated from [32]. In the example highlighted in red, $5.3 = 1.2 + 0 + 2.1 + 0 + 1.0 + 0 + 0 + 0 + 1.0$. The exact kernel values are learned during training.

The stride and size of the kernel, are hyperparameters set before training, however, the kernel values are learned during training. In computer vision and signal processing, convolutions are used to extract the relevant features and patterns in the input. Additionally, convolutions are a common form of dimensionality reduction, as the resulting convoluted matrix is smaller but still captures important information. Pooling layers can also be added after the convolutional layer which pull the most significant features from the matrix, further reducing the dimensionality [73]. These features are often then used as the input for other types of neural networks like MLPs and RNNs.

## 2.2 Models of Emotion

In this section, we give a brief overview on history of the different representations used to model emotion within the music information retrieval and music emotion recognition (MER) research community. These models discretize and quantify emotions into different sections which are argued to be fundamental emotions which describe all others. Among modern MER and linguistic literature, there are three commonly used psychological models for emotion as seen in Table 1.

| Author | Type | Year | Emotions |
|---|---|---|---|
| Russell [55] | Continuous | 1980 | Valence, Arousal, Dominance |
| Plutchik [53] | Categorical | 1980 | Fear, Anger, Joy, Sadness, Acceptance, Disgust, Expectation, Surprise |
| Ekman [21] | Categorical | 1992 | Fear, Anger, Joy, Sadness, Disgust, Surprise |

Table 1: Three different common models of emotion used in MER.

The first two models, Plutchik's and Ekman's, both create categories of emotion which they believe to be the most fundamental to human experience. In the first chapter of his 1980 book *Theories of Emotion*, Plutchik set out to define these fundamental emotions by looking at emotions from an evolutionary viewpoint and determining what advantage they would provide. He argued that a basic emotion is one in which external stimuli trigger an individual immediate response. These emotions are not synonymous with the inner state of feeling [53]. For example, a stimulus could be the threat of an enemy, which then triggers their cognition of being in danger leading to the feeling of fear. This immediate feeling drives behavior influences the organism actions, for example, running and escape.

Plutchik determined that there are four opposing pairs of basic emotions, and the varying combination of these emotions encompass all others. These emotions are Fear-Anger, Joy-Sadness, Acceptance-Disgust, and Expectation-Surprise. He also describes the associated behaviors and personality traits with them [53]. In the above example, the associated behavior of fear is to run, and a trait of an organism who is fearful would be skittish.

Developed in 1992, Ekman's six emotion model contain the same basic emotion groups as Plutchik's model, however, he excludes Acceptance and Expectation. These two emotions are excluded because Ekman argues that a basic emotion must be a distinct and brief set of neural signals such that they are distinguishable from a person's traits and personality. They must only focus on the current external stimuli, and as such, the emotion of trust is not a brief signal since it focuses on future actions. Ekman argues that Fear, Anger, Joy, Sadness, Disgust, and Surprise have nine characteristics: Distinctive universal signals, Presence in other primates, Distinctive physiology, Distinctive universals in antecedent events, Coherence among emotional response, quick onset, Brief duration, Automatic appraisal, Unbidden occurrence [21]. Additionally Ekman argues that these basic emotions are not singular emotions but encompass a family of related emotions with similar physical characteristics. However, he does not mention any computability between these groups [21].

Finally, the Valence Arousal Dominance model does not use discrete classifications to define emotion. Instead, it uses continuous values for each dimension as seen in Figure 4. These three dimensions were initially proposed in 1957 by Osgood who argued the independent combination of these three could describe any emotion. [12]. Then in [55], Russell built upon this model by connecting the three dimensions into a singular emotion space called the Circumplex Model of Emotion. In his paper, he describes how the dimensions are not independent, and how previous works like Plutchik's can be easily represented within his model. In fact, Plutchik's and Ekman's emotions only need two dimensions (Valence and Arousal) to be described. The final dominance axis has been found to be strongly correlated with valence [66], and so, many studies including ours exclude it. However, many recent studies suggest that this is one of the main sources of limitations in current work [74].

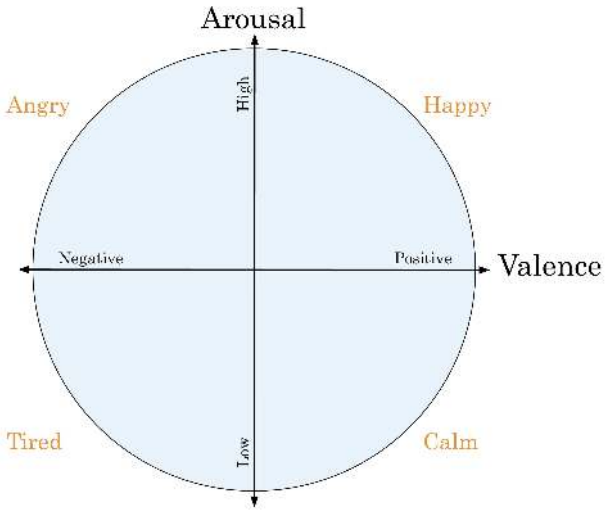



Figure 4: Russell's graph of Valence and Arousal.

## 2.3 Feature Selection Algorithms

In machine learning, more features does not always translate to better experimental performance. Large feature sets often contain redundant, correlated, and noisy information which drastically decreases model performance. The curse of dimensionality describes this concept in which higher dimensional data exponentially increases the problem space, making data points much more sparse. This effectively reduces the distances between points compared to the vast high dimensional space, causing models to be less effective as decision boundaries are less obvious. To combat this, dimensionality reduction and feature selection algorithms attempt to find the most effective, most varying, feature space. In this section we describe common feature selection and dimensionality reduction algorithms.

There are three main types of feature selection methods: *filter*, *wrapper*, and *embedded*. Filter methods use statistical measures of performance for each feature type. Features are ranked using these measures prior to model training. Common measures of performance are information, distance, dependence, consistency, similarity, and other statistical measures [65]. Instead of calculating a score for every feature, wrapper methods randomly select feature subsets to train a model, and measure the performance of that subset. The method for choosing feature subsets distinguish different wrapper algorithms. The model this method can be as simple or complex as needed, but are commonly Random Forests, Support Vector Machines, or Naïve Bayes. The final feature set is the subset which performed the best on the model. These methods are more computationally complex as they must train and test many subsets of features. Finally, embedded methods use feature selection during the model's execution. Instead of just optimizing the model's output on a loss function, a second penalty for feature amount can be added. Since these two operations are performed concurrently,

embedded methods are seen as more efficient than other methods [65].

### 2.3.1 ReliefF

One of the most common of these algorithms is known as Relief feature selection. For the purposes of this study, we choose to describe ReliefF, the multi-class version of Relief. This is a filter method in which a performance metric, or quality estimate, is calculated for each feature. This metric is calculated by choosing random instances of the dataset and computing its nearest neighbors of the same class, "nearest hits", and the nearest neighbors of each different class, "nearest misses". A hyperparameter $k$ is chosen before running to determine how many of these hits and misses to include. After finding those nearest instances, the quality estimates are updated accordingly: For nearest hits, the estimates are decreased for features that differ between instances, and, for nearest misses, it increases the estimate of features that differ between instances. Intuitively, this makes sense as features which vary between instances should be more informative if the instances are from different classes and be similar for the same class. Mathematically, this can be described as:

$$W(f) = W(f) - \mathit{diff}(A, R, H)/m + \sum_{C \neq class(R)} [P(C) \times \mathit{diff}(A, R, M(C))]/m$$

as provided by [33]. Here, $R$ is the sampled instance, $H$ is the nearest hits, $M$ is the nearest misses of class $C$, $P(C)$ is the prior probability of class $C$, and $m$ is the number of samples. The *diff* function used in [65]:

$$\mathit{diff}(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{max(A) - min(A)}$$

calculates the difference in the values of the features for instances $I_1$ and $I_2$ and normalizes them between 0 and 1. In order to compute these scores, the algorithm needs to find the $k$ nearest neighbors for every instance and update the weights for every feature leading to a time complexity of $\mathcal{O}(instances^2 \cdot features)$.

### 2.3.2 Mutual Information

Another way to quantify the performance of a feature is using mutual information. This is a concept from information theory which uses entropy ($H$) to asses how relevant a feature is in predicting a class. Shannon entropy measures how much information is gained from a measurement, or variable, that specifies a certain value [59]. In order to calculate mutual information, features, or variables in this context, are chosen iteratively and the joint probability distribution is computed between it and the target classes. The mathematical formula for mutual information is defined as $M(X, Y) = H(X) + H(Y) - H(X, Y)$ [59]. Using the continuous formula for

Shannon entropy and reducing [6] we end up with:

$$M(X,Y) = \iint p(x,y) log[\frac{p(x,y)}{p(x)p(y)}]dxdy$$

Using the joint and marginal probabilities, mutual information quantifies the shared information between the two random variables symmetrically, measuring their statistical dependence. For example, if the chosen feature $x$ had no variance across all classes, then its entropy and mutual information would be 0 since it is providing no information, and $x$ and $y$ are independent. Alternatively, if $y$ stays the same class for all values of $x$ then it would also be 0. This results in a graph of mutual information which is shaped like an upside down $U$, with the maximum mutual information value $= max(H(X), H(Y))$, meaning the values of $X$ and $Y$ are fully dependent.

### 2.3.3 L1 Regularization

Another common method of feature selection is using L1 regularization with SVMs or logistic regression. This is an embedded method which trains these models normally, however, their loss functions include L1 penalty terms. This penalty term to encourages the sum of the absolute values of the models' coefficients to be small which reduce overfitting [46]. In doing so, it forces some of these coefficient values to be 0, creating a sparse solution sets. The coefficients with value 0 are given no weight by the model, and the features associated with them are not used as their values are multiplied by 0. L2 regularization on the other hand uses the squared sum of coefficients which asymptotically approach 0. Using L1 regularization for feature selection finds these coefficients with no weight and removes their associated features entirely from the feature set. In regression problems, linear least squares regression with L1 is known as Lasso which adds an L1 penalty term $\lambda$ to the usual linear regression equation. For classification problems, logistic regression, or SVMs with linear kernels are used instead. Similar to Lasso, the L1 penalty and regularization parameter are added to logistic regression and SVM loss functions [46].

### 2.3.4 Recursive Feature Elimination

Finally, the last feature selection algorithm discussed here is Recursive Feature Elimination, or RFE. This is a wrapper method meaning it trains a model on different subsets of the features. For consistency we used logistic regression and SVM models, but other supervised models like Random Forests and Naive Bayes are also common. RFE works by iteratively removing the worst performing feature until a target amount is reached. It starts by training and evaluating the model with the entirety of the feature set, and then computing a ranking criterion for all the features. These criterion are usually the coefficient values, but they could be any other feature importance metric. The feature with the lowest ranking is removed, and then the model is re-

trained and evaluated on that new feature set [28]. Since this process is repeated until a set amount of features remain, it can be very computationally expensive, requiring many models to be trained.

### 2.3.5 PCA

Dimensionality reduction methods differ from feature selection algorithms, but they can be used to achieve the same goal. The most common dimensionality reduction algorithm is Principal Component Analysis, or PCA, which transforms high dimensional data into lower dimensions. This is done by finding linear lines or hyperplanes, known as principal components, to project data onto.

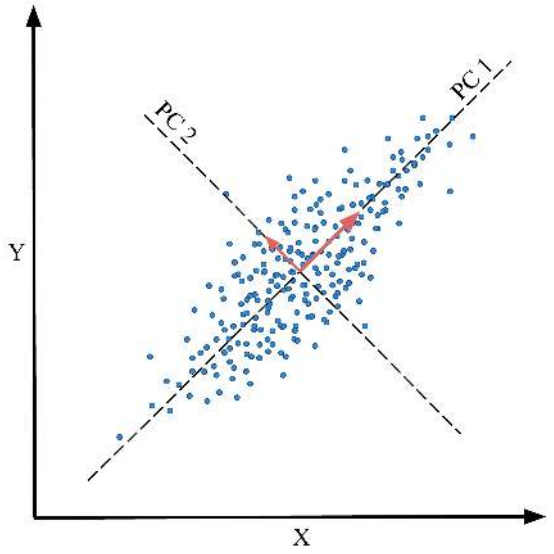



Figure 5: Visualization of two principal components in two dimensional space.

These lines maximize the directions of greatest variance in the projected data, retaining the maximum possible information of the data, but in lower dimensional spaces. An example of this can be seen in Figure 5. This is accomplished by finding the eigenvectors and eigenvalues of the data. The eigenvalues rank the amount of covariance in the data, and their corresponding eigenvectors represent the principal components themselves which the data is projected onto. The eigenvalue with the greatest value is associated with the first principal component, or the line which retains the most amount of information after reducing by one dimension. The second principal component is orthogonal to the first, and reduces by two dimensions. A hyperparameter set before reduction is chosen to determine how many dimensions to retain, however, since PCA does not account for labels, setting it too low can squish classes together.

# 3 Related Work

This section reports the history of MER up until our research, summarizing both the types of machine learning models used and modalities of input.

## 3.1 Evolution of Models

When discussing notable previous efforts in this field, the "Emotion in Music" task at MediaEval Multimedia Evaluation Campaign during 2013-2015 is important to consider [2]. For the task, the Database for Emotional Analysis of Music (DEAM) was created, and participants competed to develop the most effective models. The dataset comprised of 260 baseline low-level features extracted using OpenSMILE, with continuous ground truth labels based on the valence arousal graph. Over the three years of the competition, LSTM models consistently outperformed other approaches, including SVRs, and Multi-Linear Regression. In the final year (2015), Li et al. [39] highlighted the importance of the future context in addition to the past context of a song as equally important for classifying emotion. Their DBLSTM-ELM model demonstrated improved performance compared to other teams. Subsequently, they later improved their results by comparing different fusion models. They found that ELM, SVR, and ANN fusion consistently outperform AVG and MLR models, with SVR being particularly effective for valence and ANN for arousal [40].

Another major improvement came after Pons et al. [54] demonstrated that convolutional layers can effectively learn relevant temporal and frequency features from spectrograms. Using this, Malik et al. developed a stacked CNN-RNN model [43] using GRUs. In their experiment, their CRNN was able to achieve even better results than Li et al. [40] with 1200 fewer parameters. These results were further reinforced with work done by Yizhuo Dong et al. [19] who achieved comparable performance by using convolutional layers with the aforementioned BLSTMs on affect-salient features. However, despite these advancements, it is important to acknowledge criticism regarding the DEAM dataset, particularly concerning its labeling methods (MTurk) and lack of stratification across the four quadrants of the valence-arousal space.

During a similar time period, another music emotion recognition competition was hosted by MIREX and continued until 2020. Unlike MediaEval, the dataset was split into five discrete classification clusters instead of continuous valence-arousal values. In 2012, Renato Panda achieved the highest classification accuracy that year (67.8%) [50]. Despite continued efforts until 2020, models struggled to surpass the 68% accuracy mark on this challenging dataset, and stagnation has become an issue [57]. In response, Panda proposed a new dataset which integrated the the strengths of both competitions' datasets [49]. This dataset adopted the same taxonomy as the MediaEval but discretized the ground truth labels into the four valence-arousal quadrants. Additionally, the labels are obtained using more robust methods than DEAM. Details about this dataset are further elaborated in the next section. Panda

extracted both standard and novel (MIDI) audio features and evaluated them using a SVM, resulting in a $F_1$-Score of 76.4 with 100 features (29 novel, 71 standard) and 71.7 with 800 standard (no novel) features. These results establish a baseline we consider in our study.

## 3.2 Feature Sets

Previous research in MER has attempted to solve the issue by employing many different modalities of input. Some of these include raw audio waveforms, image spectrograms, lyrics, embeddings, music videos, and acoustic features extracted from the audio. In this section, we explore the different types of features that can be extracted and how they play a role in emotion recognition. We focus on approaches that rely on feature extraction due to their proven results. Notable examples of these studies are [42] [71] [69] [14] [49].

In [42], Lu et al. identify three feature groups, intensity, timbre, and rhythm as being emotionally relevant, and extract them from 32 millisecond audio segments. For intensity features, they calculate the spectrum sum for each frame of the clip and the spectrum distribution in each sub-band. For timbre features, they compute four spectral shape features (Brightness, bandwidth, rolloff, spectral flux) and three spectral contrast features (Sub-band peak, Sub-band Valley, Sub-band Contrast). Finally, for their rhythm features, they calculate the rhythm strength, average correlation peak, average tempo, and average onset frequency. From these thirteen features, they use a Gaussian Mixture Model to classify one of four emotions (Contentment, Depression, Exuberance, and Anxious/Frantic). With this model, they were able to achieve max classification accuracy of 86.3% using the timbre and rhythm features. However, it is important to note that this study was only conducted on classical music, and most importantly, audio segments with no inter-annotator agreement were not included. This inflated their accuracy numbers as only the easiest to classify segments were used.

Wu and Jeng use a similar approach by extracting rhythmic and timbral features, however, they expand on their feature set by including a larger variety of features [69]. These include several features from pitch, level, harmony, loudness, tonality, and Daubechies Wavelet Coefficient Histogram (DWCH). This was accomplished using Marsyas [64] and PsySound3 [9] and two other extractors DWCH and spectral contrast. In total, they computed 88 features which were then used to train a SVM. Their model was used to classify these features into eight different emotions: Sublime, Sad, Touching, Easy, Light, Happy, Exciting, and Grand. In order to obtain these classifications, they utilized the most frequent label attained by manual annotation from an online crowd sourcing survey. Finally, in order to reduce the dimensionality of their feature set, they trained their SVM individually with each feature and removed the ones with the lowest $F_1$ score. Their best performing feature set consisted of 29 features and attained a 73% cosine similarity.

Instead of a classification approach, Yang et al. used multiple linear regression, support vector regression, and BoostR models to predict valence and arousal values [71]. They used the same four extractors as Wu and Jeng, but added more features from PsySound including sharpness, volume, multiplicity, dissonance, and chord level properties for a total of 114 features. Their dataset consisted of 195 popular western, Chinese, and Japanese songs, and the ground truth labels were obtained from 253 campus volunteers who manually annotated ten randomly chosen songs. Their annotations ranged from -1 to 1 for both valence and arousal values, and the average rating for each song was calculated. Rather than greedily reducing their feature set by individually removing features, they use the ReliefF feature selection method to independently select the top five features for arousal and valence. With these, they were able to achieve their best $R^2$ values of 58.3% and 28.1% for arousal and valence respectively using the support vector machine.

More recently, Chowdhry et al. introduces mid-level features to predict the valence and arousal values for 110 movie soundtracks [14]. As opposed to the previously mentioned low-level features, which are more abstract theoretic concepts, mid-level features represent musical characteristics that are recognizable to listeners with no training. Some examples of such features include tonal and rhythmic stability, articulation, and melodiousness. Their process differs from those above as they do not compute these features directly from their audio. Instead, they train a CNN to predict the mid-level features from a spectrogram, and then use those predicted features as input to a linear regression model. With this architecture, they are able to attain correlation values of 79% and 74% for valence and arousal, respectively.

However, Panda et al. criticizes these studies stating that the lacking results and stagnation in the field are due to missing emotionally relevant features, non-uniform taxonomies, and use of private datasets [49]. For example, they reference MFCCs which were originally created for other audio tasks such as speech recognition [41], and as such, they may not contain emotional relevance. In order to solve these issues, Panda created a new dataset [49] with classes based on the quadrants Russell's valence and arousal emotional model, and conducted a comprehensive study on emotionally relevant features. In order to deem a audio feature "emotionally relevant", they extracted 1702 standard audio features from popular audio extraction software: Marsyas [64], PsySound3 [9], and MIR Toolbox [35]. In order to reduce redundancy, features with a correlation higher than 0.9, and features with 0 standard deviation were removed. Thus, 898 standard features were selected. They then added 29 high-level novel audio features which were obtained from the MIDI notes of the audio. The best results, 76.4% $F_1$-score, were obtained using a SVM on a 100 feature subset. This subset was obtained by filtering the standard features using the ReliefF method to obtain the top 71, and then combined with the 29 novel features. In comparison, the same SVM model trained on only the top 70 standard features attained a 67.5% $F_1$-score. As a result, Panda demonstrated the need for novel emotionally- and musically- relevant features.

In a survey published later, Panda et al. delve deeper into relevant audio features, trying to identify which are needed to advance the state of the art [48]. Panda et al. break up the features into eight musical dimensions and describe how they are each related to emotion: melody, harmony, rhythm, dynamics, tone color, expressivity, texture, and form. From this survey, they determined that there is not enough research on features within the musical form, texture and expressivity, and that these could help close gaps within MER. In Section 5, we describe our process of extracting our features, and the eight musical dimensions and their relation to emotion are explained.

# 4 4Q Dataset

Our work centers around comparing experimental results using the 4Q dataset created by Panda et al [49].

## 4.1 4Q

The primary dataset that Panda et al. used in their research was the 4Q Audio Dataset [49]. This dataset was developed by Panda et al. in 2018, and we will be using it to compare our results against theirs. The dataset consists of a total of 900 audio clips that are each 30 seconds in length. Their decision to create a new dataset stemmed from the issues they found with previously existing datasets, including unimpressive experimental results, missing relevant features, a lack of taxological backing in psychology, and the unavailability of private datasets.

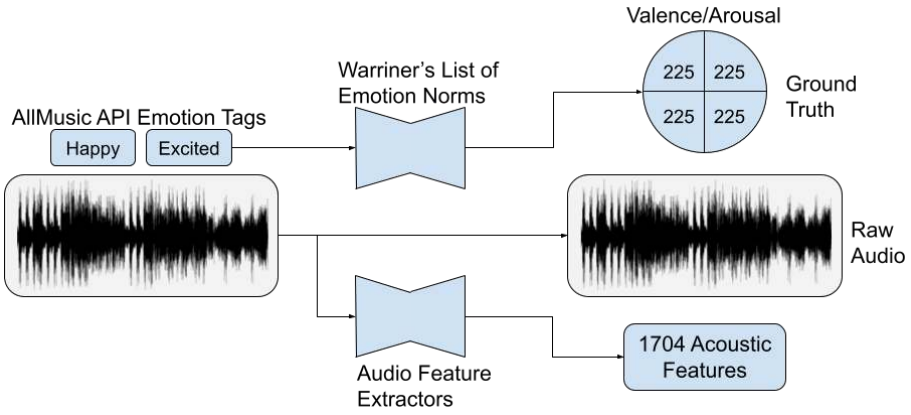



Figure 6: This flow chart represents the steps Panda et al. used to create the 4Q dataset. The results from the AllMusic API is represented on the left with the final dataset on the right.

In order to mitigate these dataset issues, the songs for the new 4Q dataset were taken from the AllMusic API and emotions were represented by 289 distinct tags. Using the Warriner list of emotion norms [66] Panda et al. was able to map the affective meaning of the tags to Russel's Circumplex Model [55]. This processes can be seen in Figure 6. The resulting emotion mapping was manually validated and songs which were not primarily dominated by one quadrant or had less than three tags were filtered out. This resulted in 900 total songs which were evenly distributed with 225 songs in each quadrant.

In addition to the audio clips, Panda et al. also provides static statistical audio features for each song. These features were extracted using a combination of Marsayas, PsySound, MIR Toolbox, and Essentia toolkits. Additionally, they extracted novel features from the MIDI notes of each audio clip after conversion. Since extracting

MIDI notes from raw audio signals is still an unsolved problem, they build upon works by Salomon et al. [56] and Dressler [20] to estimate the predominant melodies and pitch saliencies using frame and hop sizes of 1024 and 128 respectively. These values are then segmented into MIDI notes using the method proposed by Pavia et al. [47]. In this work, a series of rules are utilized to extract the notes that denote the main melodies. With these estimated MIDI notes, they perform a multitude of different computations to extract relevant features, e.g., Musical Layers Distribution. The full lists of these features can be seen in Tables 9, 10, and 12. Together with the standard statistical features, a total of 898 features were added to their dataset. They provide these features in two `CSV` files: one containing all the features, and one containing their top 100 features.

## 4.2 State of the Art

In the same paper they published their dataset, they evaluated their top 100 features on a SVM model with an RBF kernel [49]. With these features, they achieved an $F_1$-Score of 76.4%, and the confusion matrix seen in Table 2.

| | **Q1** | **Q2** | **Q3** | **Q4** | **$F_1$-Score** |
|---|---|---|---|---|---|
| **Q1** | 81.87% | 6.34% | 3.79% | 8.00% | 78.0% |
| **Q2** | 10.64% | 84.69% | 3.11% | 1.56% | 86.6% |
| **Q3** | 6.31% | 3.73% | 69.89% | 20.07% | 70.9% |
| **Q4** | 10.82% | 0.73% | 20.38% | 68.07% | 68.8% |

Table 2: This table, recreated from [49], shows the confusion matrix from their results, and the $F_1$-Score by quadrant. The matrix has been row normalized to better compare against our results.

While their features were able to classify Q1 and Q2 relatively well, Q3 and Q4 seem to be more difficult to classify. Quadrants Q3 and Q4 represent the low arousal half of the valence and arousal graph. They attribute this to more ambiguity in their dataset and labels as there was lower agreement between these two quadrants in their manual validation process. However, this issue is not unique to their study, and many other MER studies have reported issues distinguishing between valence [71] [14] [2]. Later in this paper, we will go further in depth on the process Panda et al. used, and discuss our own results using this dataset.

# 5 Feature Set Extraction

In this section, we aim to describe how we create the feature set which we later use for our experiments. We first describe the different feature extraction toolkits available to us and how we use them to extract the features, and then we describe the types of features we extract and their emotional relevance.

As seen in Section 3.2, previous studies have used Marsyas, MIR Toolbox, PsySound3, and their own extractors. We continue to use these three, however, we also include Essentia [7] and Yaafe [4] toolkits. Before using these extractors, we first mix our audio track down to a single channel and resample to 16k from 44.1k. Additionally, any audio clips that are not exactly thirty seconds long, are either clipped to this length, or padded with silence as all clips in the dataset may differ by a few hundredths of a second. These thirty seconds long clips result in a single instance having 480,000 samples. Since Panda did not provide the parameters for his extraction, we use a frame length of 2048 and a hop length of 1024. These are standard values across most of these extractors and are compatible with most of the features in these toolkits. The use of these values yield 467 time frames across the audio clip which results in a series of values for each feature. In total, we extract 710 total features from these five toolkits. We use the command line version of Marsyas and use it to extract 218 total features. We use the Python3 version of Essentia and Yaafe to extract 220 and 95 features respectively. Finally, MIR Toolbox and PsySound3 are both Matlab based, which we use to extract 100 and 77 features respectively. PsySound3 does not split the audio into frames by default, so we manually split each song into the 2048 length frames. The PsySound3 functions were then called on each of these split frames, treating them as separate audio clips, and then manually combined in post-processing using Matlab.

You will notice in the description section of the feature tables that there are annotations for certain features. These describe post-processing steps taken in order to have uniform vectors of length 467 for all features. The most common of these steps is denoted by parenthesis. This indicates that the output for that feature was a multi-dimensional array (e.g., (n, 467)). Instead of treating this as one feature, we separate them into $n$ different features. For example, the Harmonic Pitch Class Profile, or HPCP, gives a (36, 467) matrix as an output. For our feature set, they are separated into $\text{HPCP}_0$, $\text{HPCP}_2$, ..., $\text{HPCP}_{35}$ each of length 467. Another common post processing step is performed with features in which there is a singular value as an output, e.g., beats per minute. These are denoted as *Repeated* in their description, and we simply repeat the singular value 467 times. As you will see in the next paragraph, this step is not necessary but is done to simplify our workflow. Other less common techniques are described in the section in which they appear.

Once all our features are uniform vectors of length 467, we compute seven statistical features to eliminate the time dimension. These seven calculations are: mean, median, min, max, std, skew, and kurtosis. This gives us 4970 total statistical fea-

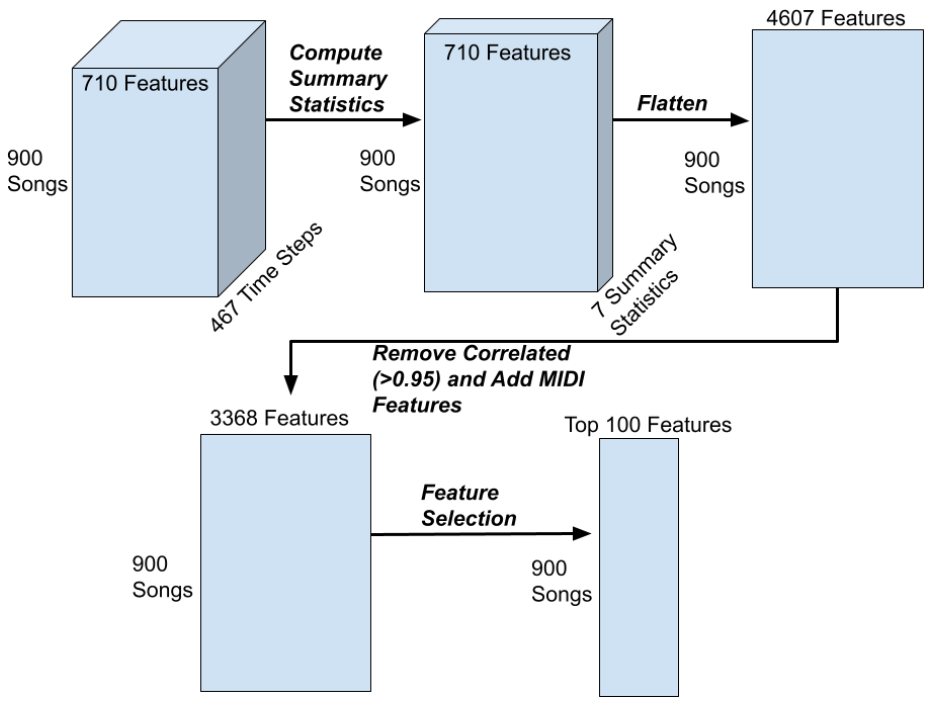



Figure 7: This flow chart shows how we reduce the dimensionality of our feature set.

tures per song, however, since some of our features like `bpm` are single values repeated, we only take the mean to get the single value, as calculating other statistics would be redundant. This results in a total of 4607 statistical features. Finally, features with a correlation higher than 0.95 across all 900 instances are added to a set, and the feature with lower mutual information is removed, leaving us with 3339 total features. We then add the 29 MIDI features obtained from Panda's et al. feature set, bringing our total to 3368 features. This process can be seen in Figure 7. Finally, reducing these features to the top 100 features using feature selection is described more in Section 6.3.

In the next section, we describe the features which make up our dataset and how they affect the emotional perception of the listener. Similar to the organization of [48], we split them into the musical dimensions: Melody, Harmony, Rhythm, Dynamics, Tone Color, Expressivity, Texture, Form, and Others.

## 5.1 Melody

Melody is defined as the successive horizontal pitches perceived by a listener as a single line, and has been described as one of the most memorable elements in a song. Melody has shown to affect a listener's emotion through pitch itself, as well as the changes in pitch. For example, Gabrielsson and Lindström found that higher pitch values, i.e. higher frequencies, have shown to lead to higher arousal energy. They

also found high ranges in pitch variation to be related to higher arousal. Together, this leads to music with high pitch and high pitch variation to be energetic songs associated with emotions like joy and anger [26]. Finally, with regards to valence, Balkwill and Thompson found that more complex melodies are associated with more negative valence values, and simpler ones with emotions like joy [5].

| Feature | Extractor | Description |
|---|---|---|
| Pitch Predominant Melody F0 | Essentia | |
| Pitch Instantaneous Confidence | Essentia | |
| Pitch Salience | Essentia | |
| Pitch Centroid | Essentia | Repeated |
| Pitch Min-To-Total | Essentia | Repeated |
| Pitch Max-To-Total | Essentia | Repeated |
| Pitch | MIR TB | (1-3) |
| Pitch Terhardt Pure Tonalness | PsySound3 | |
| Pitch Terhardt Multiplicity | PsySound3 | |
| Pitch Terhardt Chord Change Likelyhood | PsySound3 | |
| Pitch SWIPEP | PsySound3 | |
| Pitch SWIPEP strength | PsySound3 | |

Table 3: The 12 melodic features available in our five extractors.

The features we extract consist of the collection of the pitches themselves and different representations of the pitch in a song can be seen in Table 3. The more abstract features as well as any unique processing steps are described below.

#### 5.1.1 Predominant Melody F0

The Predominant Melody is an estimation of most prominent melodic line, using the YinFFT algorithm [8]. This is an optimized version of the Yin algorithm and implemented using a Hann window.

#### 5.1.2 Pitch Instantaneous Confidence

Instantaneous Confidence is a value [0-1] which describes the confidence in which the predominant melody was detected. A value of 0 means the F0 is undefined.

#### 5.1.3 Pitch Salience

Pitch Salience describes how prominent the spectral peaks are throughout time and has been used as a measure of tone sensation. Songs with several harmonics tend to have higher values.

#### 5.1.4 Pitch Centroid

Pitch Centroid is the arithmetic mean of the pitch envelope. The output is a single value.

### 5.1.5 Pitch Min-To-Total

Min-To-Total represents a ratio of the minimum pitch position to the total envelope length. The output is a single value.

### 5.1.6 Pitch Max-To-Total

Max-To-Total represents a ratio of the maximum pitch position to the total envelope length. It shows how off center the maximum amplitude is. Values close to 0 describe decrescendo and values close to 1 describe crescendo sounds. The output is a single value.

### 5.1.7 Pitch

Pitch selects the best three pitch frequencies.

### 5.1.8 Pitch Terhardt Pure Tonalness

The Terhardt algorithm for pure tonalness of pitch reflects the audibility of the spectral pitches, and is used to model pitch [63]. It extracts the virtual-pitch pattern from the spectral-pitch pattern using an algorithm was derived by Parncutt in [51].

### 5.1.9 Pitch Terhardt Multiplicity

The Terhardt algorithm for pitch multiplicity is an estimation of the number of tones simultaneously observed in a sound [63]. It similarly uses the virtual-pitch pattern derived by Parncutt in [51].

### 5.1.10 Pitch Terhardt Chord Change Likelyhood

The Terhardt algorithm for the chord change likelyhood of pitch uses the virtual-pitch pattern to determine the likelyhood of a change in the height values and was derived by Terhardt et al. [63].

### 5.1.11 SWIPEP Pitch and Pitch Strength

The SWIPEP algorithm for pitch and pitch strength estimation was developed by Arturo Camacho. It is derived similar to autocorrelation as it uses cosine as kernel to transform the audio spectrum, however, instead of the square of the magnitude of the spectrum, it uses the square root [10].

## 5.2 Harmony

Harmony, in contrast to melody, is the vertical aspect of music, or the combination of various notes played together to create different chords. Gabrielsson and Lindström found that songs with constant harmonies across the song elicit a feeling of stability, and are more associated calm and relaxing emotions, or lower arousal values. Songs with complex changing harmonies are more related to more negative anxious emotions. Similarly to melody, higher frequency chords are more associated with positive emotions happy, active and powerful. Finally, they found that changing the modality of the chords affects emotion as well. Music in the major mode usually sounds happy and fun, or high valence, whereas minor often sounds sad and serious, or low valence [26].

| Feature | Extractor | Description |
|---|---|---|
| Inharmonicity | Essentia | |
| Key | Essentia | Circle of Fifths |
| Key Scale | Essentia | Binary (Major/Minor) |
| Key Strength | Essentia | Repeated |
| Chords Change Rate | Essentia | |
| Chords Key | Essentia | Circle of Fifths |
| Chords Number Rate | Essentia | Repeated |
| Chords Progression | Essentia | Circle of Fifths |
| Chords Scale | Essentia | Binary (Major/Minor) |
| Chords Strength | Essentia | |
| Harmonic Pitch Class Profile (HPCP) | Essentia | (0-35) |
| HPCP Crest | Essentia | |
| HPCP Entropy | Essentia | |
| Key | MIR TB | |
| Key Strength | MIR TB | (2, 12) |
| Key Clarity | MIR TB | |
| Mode | MIR TB | |
| Tonal Centroid | MIR TB | (1-6) |
| Harmonic Change Detection Function | MIR TB | |
| Octive Band Signal Intensity (OBSI) | Yaafe | (0-8) |
| Mean Chroma | Marsyas | |
| Mean Average Chroma A | Marsyas | |
| Mean Min Chroma A | Marsyas | |
| Std Chroma | Marsyas | |
| Std Average Chroma A | Marsyas | |
| Std Min Chroma A | Marsyas | |
| SharpnessZ | PsySound3 | |
| SharpnessA | PsySound3 | |

Table 4: The 28 harmonic features available in our five extractors.

The Harmonic features we extract are shown in Table 4, and the more abstract features as well as any unique processing steps are described below.

### 5.2.1 Inharmonicity

Inharmonicity is the energy weighted divergence of spectral components from their closest multiple of the predominant melody F0. The values range from -1 (inharmonic) to 0 (harmonic).

### 5.2.2 Key, Chords Key, and Chords Progression

When extracted, these features are output as a string representing the key for that timestep. In order to convert these to numeric values, we use the Circle of Fifths, and represent A minor to D minor as the values 0 to 11, and C to F as the values 12 to 23.

### 5.2.3 Key Scale and Chords Scale

When extracted, these features are represented as the two strings "major" or "minor" for each timestep. These strings are converted to zero for minor and one for major.

### 5.2.4 Harmonic Pitch Class Profile (HPCP)

HPCP represents the intensities of the semitone pitch classes (A to G#).

### 5.2.5 Key Strength (MIR TB)

The MIR TB function for key strength is output as a three dimensional array with shape (2, 12, 467). In order to achieve vectors of length 467, we split by both dimensions, obtaining $\text{keystrength}_{1,1}$, $\text{keystrength}_{1,2}$, ..., $\text{keystrength}_{2,12}$

### 5.2.6 Tonal Centroid

Tonal Centroid maps the 12-bin chroma vectors to a six dimensional polytope. The pitch classes themselves are mapped to the vertices which allows the harmonic relations such as circle of fifths and minor of thirds and major of thirds appear as distances [29].

### 5.2.7 Harmonic Change Detection Function (HCDF)

HCDF is calculated as the flux of the tonal centroid vectors above [29].

### 5.2.8 Octave Band Signal Intensity (OBSI)

OBSI is calculated using triangular octave filter banks [23].

### 5.2.9 Chroma Features

Chroma Features represent the energy distribution across all pitches.

### 5.2.10 SharpnessZ

SharpnessZ is a subjective feature which measure the dullness of the audio measured in acum. One acum represents the sharpness of a band of noise centered around 1000 Hz, is one critical-bandwidth wide, and 60 dB. These are calculated using the weighed centroids of the specific loudness bands, where higher frequencies are more heavily weighted [75].

### 5.2.11 SharpnessA

SharpnessA is a revised version of SharpnessZ developed by Aures. It allows loudness to positively influence sharpness by using a different weighting function for the bands, making values more sensitive to loudness [3].

## 5.3 Rhythm

Like melody and harmony, rhythm is recognized as one of the aspects most connected with emotion, and some consider it the most important [31]. A song is composed with repetitions of notes changing over time, and each note lasting for a specific duration. Rhythm describes these repetitive patterns over time and the rate at which they occur. This rate, or tempo, is a common description for a song, and other features include how regular the tempo is or how it changes over time. When examining how rhythm affects emotion, Fernández-Soto et al. found that music with higher tempo has been linked to emotions of happiness, surprise, and higher arousal emotions. Additionally, songs with varying rhythms also illicit specific emotions [25]. This was backed up by Gabrielsson and Lindström who found that music with regular rhythms led to happy, glad, serious, and peaceful emotions. Complex and irregular rhythms on the other hand led to the emotions angry, uneasy, and amusing [26].

| Feature | Extractor | Description |
|---|---|---|
| Onsets | Essentia | Binary (1 at time of onset, 0 otherwise) |
| Oneset Rate | Essentia | Repeated |
| Beat Position | Essentia | Loudness at frame of beat, 0 otherwise |
| Beat Loudness Band Ratio | Essentia | (0-5) |
| BPM | Essentia | Repeated |
| First Peak BPM | Essentia | Repeated |
| First Peak Spread | Essentia | Repeated |
| First Peak Weight | Essentia | Repeated |
| Second Peak BPM | Essentia | Repeated |
| Second Peak Spread | Essentia | Repeated |
| Second Peak Weight | Essentia | Repeated |
| Event Density | MIR TB | |
| Tempo C | MIR TB | |
| Tempo M | MIR TB | |
| Tempo Change | MIR TB | |
| Metrical Centroid | MIR TB | |
| Pulse Clarity | MIR TB | |
| Novelty | MIR TB | |
| Fluctuation | MIR TB | (1-22) |

Table 5: The 20 rhythmic features available in our five extractors.

The rhythmic features we extract can be seen in Table 5, and the more abstract features as well as any unique processing steps are described below.

### 5.3.1 Onsets

Onsets, or events, are calculated using the peaks of the waveform to estimate the start of a note. When extracted, the function returns a list containing the time in seconds in which an onset was detected. A onset describes the beginning of a sound or note in the audio, and it is detected using an onset detection function which detects abrupt changes in the frame-to-frame energy. In order to obtain a 467 length vector, we calculate the frame which corresponds to the time of each onset, and label that frame with a one. If no onset was detected in a frame, the value is zero.

### 5.3.2 Beat Position

Beat Position is similar to the Onset function in that it returns a list of times in seconds where beats were detected, however, it also returns the loudness of those beats. Similar to how we created the vector for onsets, we calculate the corresponding frame at the time of a beat, but instead of assigning one or zero to its value, we use the loudness of the beat. If no beat was detected for a time frame, it is set to zero.

### 5.3.3 Beat Loudness Band Ratio

Beat Loudness Band Ratio is created in the same fashion as Beat Position. However, six features are created the energy in each bands in the audio. This gives more specific loudness estimates for the beats.

### 5.3.4 First and Second Peak bpm, spread, weight

These six features are calculated by creating a beats per minute histogram. The bpm and statistics are calculated for the most and second most common peaks.

### 5.3.5 Event Density

Event Density estimates the frequency of events, and is the same as onset rate.

### 5.3.6 Tempo C

Tempo C is the classic tempo calculation method in the MIR TB extractor. It is based on detecting periodicity in a range of bpms and choosing the largest score for each frame.

### 5.3.7 Tempo M

Tempo M is the metre version of tempo calculation in MIR TB. It builds a hierarchical metrical structure leading to continuous tempo curve [34].

### 5.3.8 Tempo Change

Tempo Change computes the difference between successive values of the tempo tempo curve

### 5.3.9 Metrical Centroid

Metrical Centroid assess the metrical activity in bpm. High values indicate faster elementary metrical levels predominate [34].

### 5.3.10 Pulse Clarity

Pulse Clarity is a high level feature that estimates the rhythmic clarity in a song. It can be used to describe how easily can listeners perceive the rhythmic or metrical pulsation, and it indicates the strength of beats estimated by tempo [37].

### 5.3.11 Fluctuation

Fluctuation measures the rhythmic periods and content in each bark band

## 5.4 Dynamics

The dynamics dimension refers to the loudness and energy of the notes played and how these values change over time. As you might expect, louder music is linked to higher arousal and more intense emotions like joy and anger, whereas, quieter music

is linked to calm, serene and sad music [31]. Additionally, Gabrielsson and Lindström found that more variation in the loudness can lead to negative valence emotions.

| Feature | Extractor | Description |
|---|---|---|
| Loudness | Essentia | Repeated |
| Temporal centroid | Essentia | Repeated |
| TC-To-Total | Essentia | Repeated |
| Spectral RMS Energy | Essentia | |
| RMS Energy | MIR TB | |
| Low Energy | MIR TB | |
| Loudness | Yaafe | (0-23) |
| Perceptual Sharpness | Yaafe | |
| Perceptual Spread | Yaafe | |
| Level | PsySound3 | |
| Dynamic Loudness Sharpness | PsySound3 | |
| Dynamic Loudness Average Main Loudness | PsySound3 | (1-2) |
| Dynamic Loudness Average Specific Loudness | PsySound3 | |
| Loudness | PsySound3 | |
| Average Excitation Pattern | PsySound3 | (1-3) |
| Timbral Width | PsySound3 | |
| Volume | PsySound3 | |
| Average Specific Loudness Pattern | PsySound3 | (1-3) |
| Instantaneous Loudness | PsySound3 | |
| Short Term Loudness | PsySound3 | |
| Long Term Loudness | PsySound3 | |
| Hilbert Instant Level | PsySound3 | (1-4) |
| Hilbert Instant Phase | PsySound3 | (1-5) |
| Hilbert Instant Frequency | PsySound3 | (1-4) |
| Sound Level Weighted | PsySound3 | |
| Sound Level Unweighted | PsySound3 | |

Table 6: The 25 dynamic features available in our five extractors.

The dynamic features we extract can be seen in Table 6, and the more abstract features as well as any unique processing steps are described below.

### 5.4.1 Loudness

Loudness is a measure of the subjective impression of the intensity of the sounds using Steven's power law [60].

### 5.4.2 Temporal Centroid

The Temporal Centroid represents the ratio of the envelop centroid to the envelope length.

### 5.4.3 Tc-To-Total

Tc-To-Total is similar to Temporal Centroid however, it uses the total length of the signal instead of the envelope. It is used as a measure of sound balance.

### 5.4.4 RMS Energy

The RMS Energy measures the overall amount of energy in the signal

### 5.4.5 Low Energy

Low Energy is calculated as the percentage of frames with less than average amounts of energy. It is used to show the temporal distribution of energy and how the energy changes in the song [64].

### 5.4.6 Loudness

The Yaafe toolkit implementation of Loudness calculates the loudness coefficients for each bark band normalized by the total sum. This method was proposed by Moore, Glasberg, et al. [45].

### 5.4.7 Perceptual Sharpness

Perceptual Sharpness represents the sharpness of the loudness coefficients calculated above.

### 5.4.8 Perceptual Spread

Perceptual Spread represents the spread of the loudness coefficient calculated above.

### 5.4.9 Level

Level is the unweighted height of the sound pressure in the signal.

### 5.4.10 Dynamic Loudness Average Specific Loudness

The Average Specific Loudness utilizes the dynamic loudness algorithm [11] which uses the Bark critical band rate scale to model auditory filters and auditory temporal integration.

### 5.4.11 Dynamic Loudness Sharpness

Dynamic Loudness Sharpness is an extension of the Average Specific Loudness feature in which the Zwicker and Fastl sharpness model [75] is used.

### 5.4.12 Loudness

Loudness from the PsySound3 toolkit utilizes the loudness coefficients of the ERB bands instead of the bark bands.

### 5.4.13 Average Excitation Pattern

The Average Excitation Pattern represents the degree of stimulation of auditory filters from the ERB bands prior to specific loudness transformation [45].

### 5.4.14 Timbral Width

Timbral Width is the measured width of the peaks of the specific loudness spectrum.

### 5.4.15 Volume

Volume represents the size or intensity of the sound.

### 5.4.16 Average Specific Loudness Pattern

Average Specific Loudness Pattern specifys the distribution of loudness across the bands measured in sones per ERB band, and can be used to find the loudest part of the spectrum [45].

### 5.4.17 Hilbert Instant (Level, Phase, Frequency)

These three features utilize the Hilbert Transform across the audio waveform to extract the envelope of the signal without an integration function. The magnitude of this envelope is the instant amplitude, which when transformed into decibels, is the level. Differencing it with an appropriate coefficient yields the frequency.

### 5.4.18 Sound Level Weighted

The Sound Level Weighted feature utilizes filters to emphasize certain frequencies based on equal loudness. The unweighted version of this feature uses no filters.

## 5.5 Tone Color

The definition of tone color, or timbre, begins to delve into lower level elements of music. This dimension deals with instrument identity and the properties of the unique sounds an instrument creates like the shapes of its sound waves. These low level features have been used to differentiate different sounds from one another, for example, vocals from instruments, or strings from wind instruments. While examining if an instrument conveys a specific emotion because of how it sounds or how it is used, Wu *et al.* found that instruments themselves were naturally disposed to certain emotions, e.g., "Sad and Depressed were also strongly correlated and were best evoked by the horn and flute" [68]. They also found that these emotions were strongly correlated with spectral features, especially spectral centroid and odd to even harmonic ratio.

This backs up data found by Gabrielsson and Lindström who found that instruments with sharper amplitude envelopes are linked with positive emotions while rounder ones are related to emotions like disgust and sadness. They also found that the number of harmonics also plays a role in emotion, with lower amounts being associated with emotions like boredom, happiness or sadness [26].

| Feature | Extractor | Description |
|---|---|---|
| Barkbands | Essentia | (0-26) |
| Barkband Spread | Essentia | |
| Barkband Skew | Essentia | |
| Barkband Kurtosis | Essentia | |
| High Frequency Content (Brightness) | Essentia | |
| MFCC | Essentia | (0-12) |
| GFCC | Essentia | (0-12) |
| Spectral Complexity | Essentia | |
| Spectral Centroid | Essentia | |
| Spectral Decrease | Essentia | |
| Spectral Energy | Essentia | |
| Spectral Energyband Low | Essentia | [20Hz, 150Hz] |
| Spectral Energyband Middle Low | Essentia | [150Hz, 800Hz] |
| Spectral Energyband Middle High | Essentia | [800Hz, 4kHz] |
| Spectral Energyband High | Essentia | [4kHz, 20kHz] |
| Spectral Flatness dB | Essentia | |
| Spectral Flux | Essentia | |
| Spectral Kurtosis | Essentia | |
| Spectral Spread | Essentia | |
| Spectral Skewness | Essentia | |
| Spectral Rolloff | Essentia | |
| Spectral Strongpeak | Essentia | |
| Spectral Contrast | Essentia | (0-5) |
| Spectral Valleys | Essentia | (0-5) |
| Zero Crossing Rate | Essentia | |
| Tristimulus | Essentia | (0-2) |
| Odd-To-Even Harmonic Energy Ratio | Essentia | |
| Dissonance | Essentia | |
| Melband Crest | Essentia | |
| Melband Flatness dB | Essentia | |
| Melband Kurtosis | Essentia | |
| Melband Skew | Essentia | |
| Melband Spread | Essentia | |
| Erbbands Crest | Essentia | |
| Erbbands Flatness dB | Essentia | |
| Erbbands Kurtosis | Essentia | |
| Erbbands Skew | Essentia | |
| Erbbands Spread | Essentia | |
| Spectral Entropy | Essentia | |
| Mean Power Cepstrum | Essentia | |
| Mean Real Cepstrum | Essentia | |
| Median Power Cepstrum | Essentia | |
| Median Real Cepstrum | Essentia | |
| Linear Predictive Coefficients | Essentia | (0-10) |
| LPC Reflection | Essentia | (0-10) |
| Linear Spectral Pairs | Essentia | (0-10) |
| Zero Crossing Rate | MIR TB | |
| Rolloff | MIR TB | |
| Brightness | MIR TB | |
| Spectral Flux | MIR TB | |
| Spectral Centroid | MIR TB | |
| Spectral Spread | MIR TB | |
| Spectral Skewness | MIR TB | |
| Spectral Kurtosis | MIR TB | |
| Spectral Flatness | MIR TB | |
| Spectral Entropy | MIR TB | |
| Spectral Irregularity | MIR TB | |
| MFCC | MIR TB | (1-13) |
| Roughness | MIR TB | |
| Spectral Crest Factor | Yaafe | (0-18) |
| Spectral Decrease | Yaafe | |
| Spectral Flatness | Yaafe | |
| Spectral Flatness Per Band | Yaafe | (0-18) |
| Spectral Slope | Yaafe | |
| Amplitude Modulation | Yaafe | 0-7 |
| Envelope | Yaafe | 0-10 |

Table 7: 66 out of 112 timbre features available in our five extractors (part 1 of 2).

| Feature | Extractor | Description |
|---|---|---|
| Spectral Centroid 1st Moment | PsySound3 | |
| Spectral 2nd Moment | PsySound3 | |
| Spectral 3rd Moment | PsySound3 | |
| Spectral 4th Moment | PsySound3 | |
| Spectral Std | PsySound3 | |
| Spectral Skewness | PsySound3 | |
| Spectral Kurtosis | PsySound3 | |
| Average Power Spectrum | PsySound3 | (1-5) |
| Cepstral Real Centroid 1st Moment | PsySound3 | |
| Cepstral Real 2nd Moment | PsySound3 | |
| Cepstral Real 3rd Moment | PsySound3 | |
| Cepstral Real 4th Moment | PsySound3 | |
| Cepstral Real Std | PsySound3 | |
| Cepstral Real Skewness | PsySound3 | |
| Cepstral Complex Centroid 1st Momement | PsySound3 | |
| Cepstral Complex 2nd Moment | PsySound3 | |
| Cepstral Complex 3rd Moment | PsySound3 | |
| Cepstral Complex 4th Moment | PsySound3 | |
| Cepstral Complex Std | PsySound3 | |
| Cepstral Complex Skewness | PsySound3 | |
| Average Gamnitude Cepstreal Real | PsySound3 | (1-2) |
| Tonal Dissonance HK | PsySound3 | |
| Tonal Dissonance S | PsySound3 | |
| Spectral Dissonance HK | PsySound3 | |
| Spectral Dissonance S | PsySound3 | |
| Roughness | PsySound3 | |
| Average Roughness | PsySound3 | (1-4) |
| Mean Zero Crossing | Marsyas | |
| Mean Centroid | Marsyas | |
| Mean Rolloff | Marsyas | |
| Mean Flux | Marsyas | |
| Mean MFCC | Marsyas | (1-13) |
| Mean SCF | Marsyas | (1-24) |
| Mean SFM | Marsyas | (1-24) |
| Mean LSP | Marsyas | (1-18) |
| Mean LPCC | Marsyas | (1-12) |
| Std Zero Crossing | Marsyas | |
| Std Centroid | Marsyas | |
| Std Rolloff | Marsyas | |
| Std Flux | Marsyas | |
| Std MFCC | Marsyas | (0-12) |
| Std SCF | Marsyas | (1-24) |
| Std SFM | Marsyas | (1-24) |
| Std LSP | Marsyas | (1-18) |
| Std LPCC | Marsyas | (1-12) |

Table 8: 46 out of 112 timbre features available in our five extractors (part 2 of 2).

The tone color features we extract can be seen in Table 7 and 8, and the more abstract features as well as any unique processing steps are described below.

### 5.5.1 Spectral EnergyBand Low

Spectral EnergyBand Low contains the energy levels in the range 20 Hz to 150 Hz.

### 5.5.2 Spectral EnergyBand Middle Low

Spectral EnergyBand Middle Low contains the energy levels in the range 150 Hz to 800 Hz.

### 5.5.3 Spectral EnergyBand Middle High

Spectral EnergyBand Middle High contains the energy levels in the range 800 Hz to 4k Hz.

### 5.5.4 Spectral EnergyBand High

Spectral EnergyBand High contains the energy levels in the range 4k Hz to 20k Hz.

### 5.5.5 Spectral Flatness dB

Spectral Flatness dB is a measurement of the flatness of the spectrum in the decibel scale. It is used as a measure of if the audio is more tone-like or noisy. High values indicates that the spectrum has the same power in all bands, i.e. white noise, whereas low values means the power is concentrated in smaller amount of bands.

### 5.5.6 Spectral Flux

Spectral Flux measures the distance between successive spectral frames.

### 5.5.7 Spectral Rolloff

Spectral Rolloff measures the amount of high frequency energy up to a frequency threshold determined by the total energy [64].

### 5.5.8 Spectral Strongpeak

Spectral Strongpeak is the ratio between the magnitude of the max peak in the spectrum and width of that peak in log10-frequency. It reveals how pronounced the maximum peaks are from the rest.

### 5.5.9 Zero Crossing Rate

Zero Crossing Rate is the amount of sign changes between consecutive values over the total number of values. Noisy audio tends to have higher values.

### 5.5.10 Tristimulus

Tristimulus measures the mixture of harmonics in three separate sections. The first section measures the first harmonic, the second section is the third and fourth harmonics, and the last measures the rest.

### 5.5.11 Odd-To-Even Harmonic Energy Ratio

Odd-To-Even Harmonic Energy Ratio calculates the ratio of the odd, e.g., clarinet, and even, e.g., trumpet, harmonic energy.

### 5.5.12 Mel Bands

The Mel Bands contain the energy of a spectrum in the mel scale. This scale is obtained using triangular frequency-domain filters.

### 5.5.13 Equivalent Rectangular Bandwidth (ERB) Bands

ERB bands contains the energy of the spectrum using the ERB scale. This scale is used to closer approximate human hearing, and is computed by filtering using rectangular band-pass filters [44].

### 5.5.14 GFCC

GFCCs, or gammatone feature cepstrum coefficients, are analogous to MFCCs but using the ERB bands.

### 5.5.15 Spectral Centroid

The Spectral Centroid is the first moment of the spectrum, and measures the brightness of the audio.

### 5.5.16 Spectral spread

The Spectral spread is the second moment of the spectrum, and measures its dispersion.

### 5.5.17 Spectral skewness

The Spectral skewness is the third moment of the spectrum, and measures its symmetry.

### 5.5.18 Spectral Kurtosis

The Spectral Kurtosis is the fourth moment of the spectrum, and measures its peakedness.

### 5.5.19 Spectral Irregularity

Spectral Irregularity represents the variation of successive peaks in the spectrum.

### 5.5.20 Spectral Crest Factor Per Band (SCF)

SCF represents the peakedness of the spectrum per log spaced bands of 1/4 octaves.

### 5.5.21 Spectral Decrease

Spectral Decrease represents the amount of of negative slope in the spectral amplitude [52].

### 5.5.22 Spectral Flatness

Spectral Flatness calculates the ratio between the geometric and arithmetic mean.

### 5.5.23 Spectral Flatness Per Band (SFM)

SFM is the inverse of the Spectral Crest Factor.

### 5.5.24 Spectral Slope

Spectral Slope is calculated using linear regression on the spectral amplitude.

### 5.5.25 Amplitude Modulation

The Amplitude Modulation contains measurements of tremolo and grain [22].

### 5.5.26 Cepstrum

Cepstrum features like *Cepstral Real Centroid 1st Moment* are statistical features over the time domain of the cepstrum. The cepstrum is the inverse Fourier transform of the log spectrum, or the log complex spectrum.

## 5.6 Expressivity

The next three dimensions, Expressivity, Texture, and Form are more complex descriptions of music, and contain less standard features to be extracted. Instead, many of these features come from the MIDI notes, and are provided by Panda. Additionally, fewer studies have explored the connection between these dimensions and emotion.

Expressivity is a musical dimension which describes the way rhythm, dynamics, and articulation are used in conjunction. Staccato articulation is associated with higher intensity and arousal [26].

| Feature | Extractor | Description |
|---|---|---|
| Silence Rate | Essentia | 20 dB |
| Silence Rate | Essentia | 30 dB |
| Silence Rate | Essentia | 60 dB |
| Silence Rate | Essentia | 90 dB |
| Vibrato Frequency | Essentia | |
| Vibrato Extend | Essentia | |
| Average Silence Rate Low Energy | MIR TB | |
| ORIGINAL EXPRESSIVE TECHNIQUES Tremolo Notes in Cents (Mean) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Extent (Std) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Tremolo Higher Notes Coverage (C4) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Tremolo Notes in Cents (Max) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Rate (Kurtosis) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Base Freq (Kurtosis) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato to Non Vibrato Notes Ratio | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Base Freq (Min) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Length (Kurtosis) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Rate (Std) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Base Freq (Std) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Extent (Kurtosis) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Glissando Direction (Up) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Length (Skewness) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES ART-Other Notes Duration (Mean) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Glissando Extent (Std) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Tremolo Notes in Cents (Min) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Vibrato Higher Notes Coverage (C4) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Glissando Length (Std) | [49] | |
| ORIGINAL EXPRESSIVE TECHNIQUES Glissando Slope (Std) | [49] | |

Table 9: The 27 expressive features available in our five extractors.

The expressive features we extract, as well as the novel MIDI features from Panda, can be seen in Table 9, and the more abstract features are described below.

### 5.6.1 Silence Rate

The Silence Rate determines if frames contain instant power levels which are below certain thresholds. We use 20, 30, 60, and 90 decibels as thresholds.

### 5.6.2 Vibrato Frequency

Vibrato Frequency is an estimate of the speed of the vibrato present in the audio in hertz.

### 5.6.3 Vibrato Extend

Vibrato Extend estimates the depth (peak to peak) of the vibrato in cents. Both Vibrato Extend and Vibrato Frequency use the pitch contour to determine the presence of vibrato.

### 5.6.4 Average Silence Ratio Low Energy

Average Silence Ratio Low Energy is determined the same way as RMS Energy but without the square root. It is a measure of silence in frames to model articulation. A frame is considered silent if the energy is lower than 0.5. Low values mean fewer silent frames, i.e., legato, and higher values indicate staccato [24].

## 5.7 Texture

Texture describes the way rhythm, melody, and harmonic information is combined in music, and it has no standard available audio features. There have been few studies on its impact on emotion, however, Webster and Weir demonstrated it can influence emotion by altering other concepts like tempo and mode [67]. These novel features can be seen in Table 10.

| Feature | Extractor | Description |
|---|---|---|
| ORIGINAL TEXTURE Musical Layers (Mean) | [49] | |
| ORIGINAL TEXTURE Musical Layers (Std) | [49] | |
| ORIGINAL TEXTURE State Transitions ML1 - ML0 (Per Sec) | [49] | |
| VOICE TEXTURE State Transitions ML0 - ML1 (Per Sec) | [49] | |
| ORIGINAL TEXTURE Musical Layers (Max) | [49] | |
| ORIGINAL TEXTURE ML1 Monophonic Texture (Percentage) | [49] | |
| ORIGINAL TEXTURE State Transitions ML2 - ML3 (Per Sec) | [49] | |
| ORIGINAL TEXTURE State Transitions ML1 - ML2 (Per Sec) | [49] | |

Table 10: The 8 texture features obtained from Panda.

## 5.8 Form

Form considers the composition and structure of sections of a song. It describes the layout of a piece in sections, or frames. Like texture, few studies have examined how form affects emotion, but Balkwill and Thompson found that less complex forms resulted in more positive emotions, like joy or peace. Contrarily, more complex forms led to negative emotions which can vary in arousal values depending on the dynamics [5].

| Feature | Extractor | Description |
|---|---|---|
| Similarity Matrix | MIR TB | |
| Novelty Curve | MIR TB | |

Table 11: The 2 form features available in our five extractors.

The two form features we extract, can be seen in Table 11, and are described below.

### 5.8.1 Similarity Matrix

The Similarity Matrix contains the correlation values of between all the frames of the audio.

### 5.8.2 Novelty Curve

The Novelty Curve represents the temporal probability of the presence of transitions between states, or peaks, and their importance, or peak height [36].

## 5.9 High level and Voice

Finally, we include some high level features which do not conform to any of the above dimensions. These can be seen in Table 12.

| Feature | Extractor | Description |
|---|---|---|
| Danceability | Essentia | |
| ORIGINAL VAT Probability of Creaky Voice (Skewness) | [49] | |

Table 12: The 2 high-level features available in our five extractors.

### 5.9.1 Danceability

The Danceability feature estimates the danceability of a song on a scale of zero to three, where larger numbers are more danceable. This is done using the Detrended Fluctuation Analysis (DFA) algorithm [62].

From all these studies, it is evident the emotion of a song is not determined by one single musical dimension. Music is complex and there is not a one size fits all approach to MER. These dimensions are not independent, and a change in one dimension can affect the other.

# 6 Experiments

In this section we discuss the use the features we extracted on the 4Q dataset and compare their performance against the features Panda et al. extracted. We start by replicating their study using the features made available from [49], and then we evaluate our features using a multitude of different machine learning models. In order to compare our results, we calculate the $F_1$-score for our predictions, which is defined as:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

We use the macro average calculation for $F_1$ as there is no class imbalance in the 4Q dataset. Finally, for consistency and comparable results, we use ten-fold cross validation across twenty independent runs for all of our experiments. When creating the folds, we stratify the classes across each fold, ensuring each fold has a similar number of examples for each class.

## 6.1 Replication Study

In order to create a baseline against which we compare our results, we first replicate the experiment Panda et al. performed [49]. In their experiment, they used the 4Q dataset they created, and the top 100 statistical features they extracted using MIR TB [35], Marsyas [64] PsySound3 [9], and the novel MIDI features they created. This feature set consisted of 71 standard features and 29 novel features. This subset was attained by reducing their 1702 standard extracted features using ReliefF feature selection and combining the top 71 with the novel features as described in Section 3.2. For their model, the authors chose to use a SVM with a RBF kernel function. Employing ten-fold cross validation across twenty independent runs, they were able to achieve a $F_1$-score of 76.4%.

In order to replicate this study, we use the top 100 feature set provided by Panda et al. [1]. We create our own ensemble of SVM classifiers using bagging to reduce variance. We then standardize the features such that the mean and standard deviation for each independent feature is zero and one respectively. Similarly to their study, we employ ten-fold cross validation across 20 independent runs, and using their provided feature set, we achieve a $F_1$-score of 76.8%. Interestingly, this is a slightly higher result than Panda et al. reported in their study [49]. This difference may be due to our use of an ensemble model instead of a single SVM model. Nevertheless, in our replication

[1] https://mir.dei.uc.pt/downloads.html

study we achieve comparable results to their reported $F_1$-score, and we will use our 76.8% result as a baseline moving forward.

## 6.2 Other Inputs

For the purpose of demonstrating that music emotion recognition is a difficult task that necessitates feature extraction, we first use raw representations of the data in order to train deep learning models. These representations consist of the raw audio waveform, the mel-spectrogram, and MFCCs. Since feature extraction is a complicated and time-consuming step, these representations have the advantage of being fast to generate and easy to interpret.

Raw audio waveforms require no transformation of the data as they are the amplitude of the audio over time. It is the most primitive type of audio input, but it is highly complex. Since this representation changes over time, we use it to train a Bidirectional LSTM, or BLSTM, to classify emotion. In order to use our waveforms, we first mix down our audio track to a single channel and downsample to 16k from 44.1k. Additionally, any audio clips that are not exactly thirty seconds long, are either clipped to this length, or padded with silence. This is necessary as the clips in the dataset often differ by a few hundredths of a second. These thirty seconds long clips result in a single instance containing 480,000 samples.

For our BLSTM, we use three BLSTM layers with dropout in between followed by two fully connected layers. We use mean squared error and train for 100 epoch with a learning rate of 0.001 and the Adam optimizer. Given the results shown in Table 13, we observe the complexity of the waveform representation makes it very difficult to obtain relevant information from, and the model's predictions barely beat random choice.

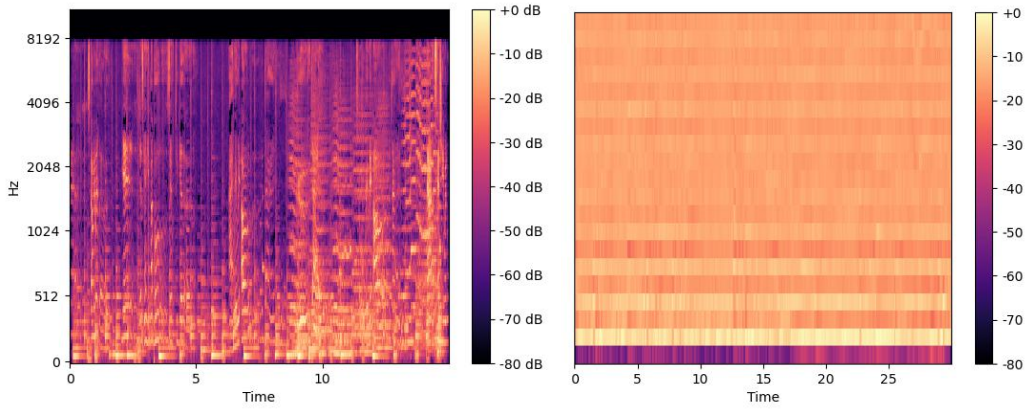



Figure 8: An example of a Mel spectrogram (left) and MFCC (right) from the dataset.

Spectrograms are visual representations of audio that show frequency changes over time by taking many separate Discrete Fourier Transforms. Their frequencies

| Input Type | Model | $F_1$ Score |
|---|---|---|
| Waveform | BLSTM | 33.2 |
| Mel spectrogram | CNN | 54.6 |
| Mel spectrogram | CRNN | 37.7 |
| MFCCs | CNN | 48.4 |
| MFCCs | CRNN | 43.8 |

Table 13: $F_1$-scores for waveform, spectrogram, and MFCC representations of audio.

and amplitude are then stacked to be visualized across time. Mel spectrograms are a special kind of spectrogram which change the scale of the frequency axis. After computing DFT of a slice of the audio, the output is sent through a mel filter bank which converts it to the mel scale. Mel spectrograms have been proven to be more useful in machine learning as the mel scale is more similar to human perception of sound, as humans are much more adept at discriminating lower frequency sounds than high ones [61].

MFCCs, or Mel-frequency Cepstral Coefficients, are a set of coefficients that represent the short term power spectrum of an audio clip. Similarly to the mel spectrograms, the DFT and mel filter banks are used to obtain the mel scale frequencies over time. Finally, in order to obtain the coefficients, a second DFT is applied to the mel spectrum [17].

In order to obtain these representations, we employ the librosa python library to extract and save them as `jpeg` files. We then crop each image to a height of 370 and width of 496 pixels to remove the white border around each image. Subsequently, we construct a convolutional neural network to see if emotional information can be extracted from these visuals. Our model has three convolutional layers with increasing filter amounts, each with 3 by 3 kernels. In between each of these layers we include max pooling and dropout layers. Finally, we flatten the output from these layers and input them into two fully connected layers for classification. For consistency, we use the same model for both mel spectrograms and MFCCs, and the results can be seen in Table 13.

## 6.3 Feature Selection

Our feature set consists of over 3339 statistical features for each audio instance. The details for their extraction can be seen in Section 5. In order to compare our feature set against Panda et al., we use feature selection to obtain the top 100 features for our feature set. This section details how we reduce our large feature set into different subsets.

First, since we were unable to extract the novel MIDI features detailed in [49], we add these features to our total initial feature set before performing feature selection. This brings our total feature amount up to 3368. Adding these MIDI features before feature selection allows these features to removed during selection, a process similar

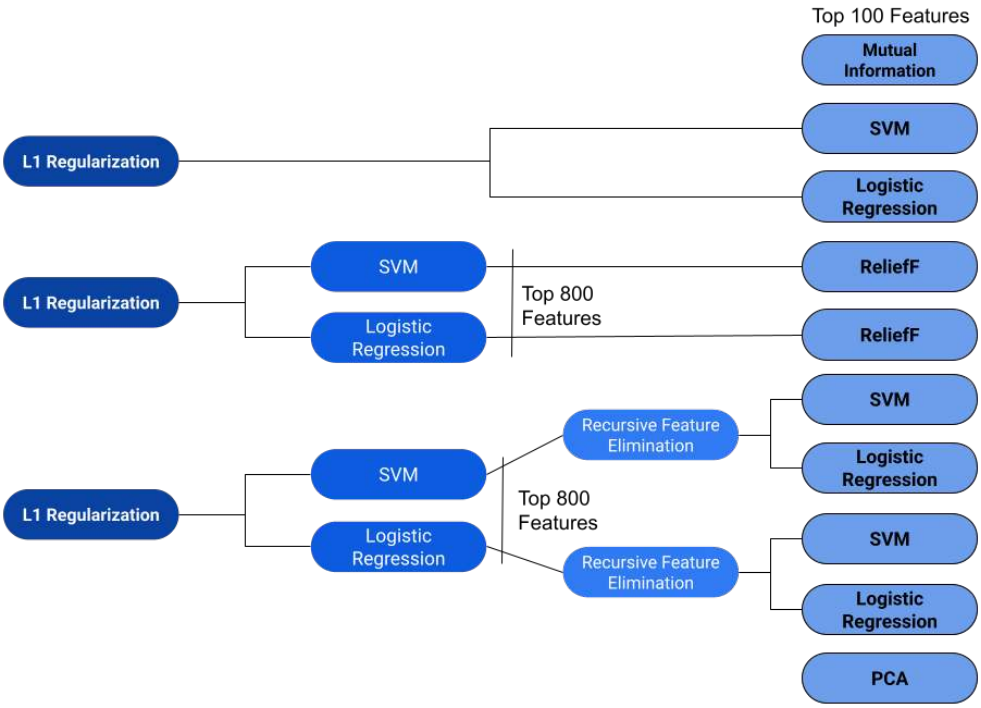



Figure 9: The ten feature subsets we obtain from feature selection and dimensionality reduction methods.

to the one done in [49].

Next we use the feature selection algorithms mentioned in Section 2.3 to create nine separate feature sets of 100 total features. The first three of these sets are created with mutual information, L1 regularization using a logistic regression model, and L1 regularization using a SVM model. These three methods are the fastest options computationally that we use, and are able to process the 3368 total features in a reasonable amount of time. The next six sets are created using ReliefF or RFE methods. Since these are more computationally complex methods and depend on feature amount, we use L1 regularization to first reduce to 800 features (about 16% of the original feature set). For this step, both the logistic regression and SVM models are tested for L1 regularization. The ReliefF and RFE methods are then performed on these smaller 800 features subset. Additionally, since RFE itself uses different models for performance, we again use logistic regression and SVM models. Finally, we also include Principal Component Analysis, or PCA. This is no a feature selection algorithm and instead a dimensionality reduction method, however, it yields the same outcome we are trying to achieve. This leads to the ten feature subsets seen in Figure 9.

## 6.4 Models Based on Features

Using the same approach as we describe in Section 6.1, we use an ensemble of SVMs to generate classifications. However, instead of using the 100 features provided by Panda et al., we use the 9 feature sets described above. An additional experiment is done using Principal Component Analysis for dimensionality reduction on all 3368 features instead of feature selection. For similar comparability, we use 10 fold cross validation across 20 independent runs, and measure the performance using macro $F_1$-score.

| **Feature Selection Algorithm** | **$F_1$-Score** |
|---|---|
| Mutual Information | 62.04 |
| L1 Regularization (SVM) | 75.5 |
| L1 Regularization (logistic regression) | 75.6 |
| L1 Regularization (SVM) + ReliefF | 70.74 |
| L1 Regularization (logistic regression) + ReliefF | 67.77 |
| L1 Regularization (SVM) + RFE (SVM) | **78.57** |
| L1 Regularization (SVM) + RFE (logistic regression) | 78.54 |
| L1 Regularization (logistic regression) + RFE (SVM) | 77.77 |
| L1 Regularization (logistic regression) + RFE (logistic regression) | 77.92 |
| PCA (100) | 68.58 |
| Panda et al. [49] | 76.4 |
| Replication Study (Section 6.1) | 76.8 |

Table 14: $F_1$-Scores obtained for our SVM model on different feature sets all with 100 features.

From these results, we can conclude that the feature set obtained using L1 regularization with an SVM and Recursive Feature Elimination with an SVM resulted in the best feature set, as seen in Table 14. Together, this feature set contains more emotionally relevant information than the one Panda et al. used. Despite using ReliefF for their feature selection, we observed noticeably worse results using ReliefF compared to RFE, further emphasizing the differences in our dataset. Moving forward, we use Panda's et al. top 100 features and our top generated feature set to compare our features on different machine learning models.

Using these two datasets, we use a random forest for classification instead of an SVM. As described in Section 2.3, a random forest is an ensemble of decision trees, where each tree is trained with a random subset of features and instances. The max depth hyperparameter limits how deep a tree can grow in which low numbers tend reduce overfitting but may model a less expressive function. Another hyperparameter, the number of estimators determines the amount of decision trees in the forest. Through our preliminary testing, we found that a max depth of 25 and 200 estimators to perform well. However, we did not perform comprehensive hyperparameter search for any of the machine learning models in this section. As seen in Table 15, our feature set again outperforms Panda et al.

Finally, we train a fully connected multilayer perceptron neural network. Instead of splitting our folds into 9:1 training to test ratios, we include a validation set,

splitting our folds into 8:1:1 train to test to validation. This validation set is held out during training, and evaluated after every epoch. This allows the model to learn from new data which it is not trained on, but still make adjustments to the parameters based on its performance. Additionally, instead of standardizing our features such that the mean and standard deviation for each independent feature are zero and one respectively, we normalize them between the range zero and one. Our MLP model consists of four hidden layers each with a depth of 256 with dropout layers in between each. For our hidden layers, `tanh` activation functions are applied, and our output layer uses a `softmax` function. Finally, we train our model with categorical cross entropy as our loss function and use the Adam optimizer algorithm with a learning rate of 0.00001. We train for 250 epochs resulting in an $F_1$-score of 79.89 as seen in Table 15.

| **Model** | **Panda's et al. Feature Set** | **Our Feature Set** |
|---|---|---|
| SVM | 76.8 | 78.57 |
| Random Forest | 71.7 | 73.1 |
| MLP | 74.8 | **79.89** |

Table 15: A comparison of $F_1$-scores between the feature set Panda et al. extracted versus the feature set we extracted, across multiple different machine learning models.

It is evident from these results that models trained with our feature set outperforms Panda et al. across all three models, perhaps indicating that it contains more emotionally relevant features. Interestingly, the MLP model outperforms the SVM on our dataset, but it fails to produce similar results on Panda et al. This may indicate that our feature set contains more complex relationships that benefit from more complex models. In order to prove this, we test different feature sizes to see how many more relevant features our set contains. Using L1 Regularization and RFE feature selection which produced the best results in Table 14, we produce subsets of our features with sizes: 70, 150, 200, 300, and 500. These sets are then evaluated using the same SVM and MLP models as above.

| **Feature Size** | **SVM** | **MLP** |
|---|---|---|
| 70 | 78.04 | 78.20 |
| 100 | 78.57 | 79.89 |
| 150 | 79.78 | 80.58 |
| 200 | 79.51 | 81.51 |
| 300 | 77.64 | **82.03** |
| 500 | 76.45 | 81.20 |
| Panda et al. 100 [49] | 76.8 | 74.8 |

Table 16: $F_1$-Scores comparing different feature set sizes evaluated on SVM and MLP. All sets were obtained using the best performing method from Table 14

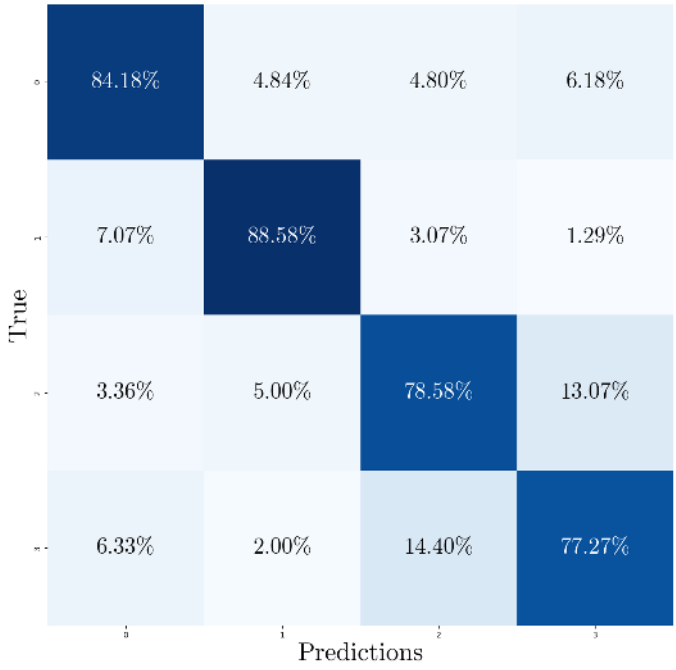



Figure 10: The confusion matrix for 300 features evaluated on an MLP.

From the results in Table 16 and the confusion matrix in Figure 10, it is clear that our MLP model is able to learn the more complex relations of our feature set. It is important to note the increase in accuracy in Q3 and Q4 when compared to Panda's et al. in Table 2. Our feature set is better able to distinguish between valence in the low arousal space. Finally, we argue that our overarching feature set contains more emotionally relevant information than Panda's et al. set. This is evidenced by the fact that the MLP model underperformed the SVM model on their feature set.

# 7 Discussion

Our experiments consisted of a comparison of the features we extracted to the features Panda et al. extracted. This comparison continues a scholarly discussion of relevant features, which began with [42] in which they used 13 intensity, timbre, and rhythm features to classify emotion in classical music. Since then, the amount and types of features used by researchers has increased, generally increasing performance. In our work, we further investigate emotionally relevant features to predict the quadrant of Russell's valence and arousal model.

When comparing our base 100 feature subset and Panda's et al. top 100 features, there are many differences. Firstly, the inclusion of the Essentia and Yaafe feature extractors proved to increase the amount of new types of features. This can be seen in Table 17. This table counts the amount of novel features in our set that were not present in Pandas et al. It is evident that the addition of Essentia introduced new feature types which were not in Panda's et al original feature set. These features better represented the emotional space of the audio and outperformed other features from Marsyas, MIR TB, and PsySound3. Additionally, the amount of novel features chosen by our feature selection was lower than in Panda's et al. This is important as it raises the question of the importance of these novel features.

| **Toolbox** | **Count** |
|---|---|
| Essentia | 24 |
| MIR ToolBox | 11 |
| PsySound3 | 2 |
| Yaafe | 2 |
| Marsyas | 3 |
| Novel | -18 |

Table 17: Number of feature types by toolbox that were present in our top 100 but not in Panda's et al. feature set.

When scaling our dataset to features amounts as seen in Table 16, our feature set outperforms Panda et al., even with 30 fewer features. This demonstrates the robustness of our most important features. When scaling up the amount, the complexity of the feature set increases as well, however, we show that a more complex model like a MLP is able to make better predictions from this. We observe the best results from 300 total features from an MLP, outperforming Panda et al. This shows that the 100 features is not a limit, and more dimensions are still emotionally relevant.

| Musical Dimension | 300 Ours | 100 Ours | Panda et al. |
|---|---|---|---|
| Melody | 16 | 4 | 1 |
| Harmony | 49 | 16 | 6 |
| Rhythm | 20 | 4 | 4 |
| Dynamics | 12 | 4 | 4 |
| Tone Color (Timbre) | 182 | 61 | 56 |
| Expressivity | 13 | 8 | 20 |
| Texture | 4 | 2 | 8 |
| Voice | 1 | 1 | 1 |
| High-Level | 3 | 0 | 0 |
| Total | 300 | 100 | 100 |

Table 19: The distribution of features within the musical dimensions for our datasets and Panda's et al.

| Toolbox | 300 Ours | 100 Ours | Panda et al. |
|---|---|---|---|
| Essentia | 111 | 33 | 0 |
| MIR ToolBox | 48 | 17 | 17 |
| PsySound3 | 16 | 6 | 16 |
| Yaafe | 30 | 12 | 0 |
| Marsyas | 79 | 21 | 38 |
| Novel | 16 | 11 | 29 |
| Total | 300 | 100 | 100 |

Table 18: Toolbox distribution for our 100 and 300 feature subsets and Panda's et al. feature set

From Table 18 we can see, that when scaling to 300 features, Essentia, MIR TB, Yaafe, and Marsyas scale by roughly three as expected, however, Psysound3 is scaled by 8 and the novel features by only about 1.5. For Psysound3, this indicates that it does not have the top end strongest features, but still contains useful features. For the novel features, this indicates that almost half of the 29 features are not as useful. We also find that Essentia features dominate the distribution with more than a third of the total features coming from the extractor. A similar scaling relationship can be seen when viewing the features by musical dimension as seen in Table 19. Some of the more major discrepancies in the musical dimensions in our feature sets versus Panda et al. is the amount of Expressivity and Harmony features.

Finally, we can see that Timbre features dominate the majority of the features in both our and Panda's et al. feature sets. This is congruent with the findings of [42] and [49]. The Expressivity, Texture, and Voice dimensions mostly contain novel features which continues to show that not all of these features are necessary for emotional representation.

## 7.1 Limitations

Our analysis of our feature set compared to Panda et al. was only trained on the 4Q dataset used by Panda et al. themselves in [49]. It is important to note that

this dataset only contained 900 total songs, and these songs are distributed evenly across the four quadrant classes. Additionally, the ground truth labels were obtained using emotional tags mapped to the quadrants. While we outperform Panda et al. on this dataset, further investigations are necessary. It is possible that a larger, imbalanced, more noisy dataset with labels obtained from crowd sourced methods might not produce the same results. The features set we extract could be overfit to this dataset, and would not attain the same performance on other datasets. We do not believe this to be the case as our set with 300 features outperformed our set with 100, improving generalization, however, this cannot be proven until more datasets are tested.

The same overfitting argument could be made about hyperparameter tuning as well. While we did not over tune these parameters and instead chose values which performed well through our experiments, there is a possibility that the values we chose for our parameters were more optimal for our features, as opposed to the information gained from the features themselves.

Additionally, the low number of total songs in the dataset also limits the ability to train larger and more advanced machine learning models. This is because these models require an immense amount of training data to tune their parameters which is not possible if the dataset is only 900 instances.

Since we were unable to extract our own MIDI representations from the audio signals, we could not extract our own novel features. Instead we used the ones provided by Panda et al. [49]. Although some of these features were eventually filtered out by the feature selection algorithm, not having these features limits our potential to run this same experiment on other datasets.

A main limitation with our method is the time necessary to extract the necessary features from the audio. This limits real time applications in which a user would input a song and expect a classification within seconds. In reality, this would take a minute or two to compute the features.

Finally, the ground truth labels, the four quadrants of Russell's valence and arousal model, are purposefully chosen as an easier task. Performance on other labeling methods such as a regression task [71] or a more complex tagging system with multiple possible tags [38] needs to be studied.

## 7.2 Future Work

The most obvious focus of our future work would be the consideration of more datasets that address the limitations described above. An obvious choice would be the DEAM dataset [2]. This dataset contains major class imbalance, many more audio instances, and most importantly is a regression problem within the valence and arousal space. With this dataset, we would need to conduct the same study, extract the necessary features, perform feature selection, and evaluate these features with different models. An interesting comparison would be to see how similar the feature sets between the

two datasets are after feature selection, and how each feature set performs on across both datasets. The increased amount of songs in DEAM would also allow the use of more advanced models which require more training instances, and allow us to perform a more comprehensive hyperparameter search. Utilizing this method would hopefully result in better generalization across datasets and reinforce our findings.

As stated in the introduction of this paper, we restricted our features to those which describe the audio and not the lyrics and vocals. Since lyrics and vocals are an important aspect of music, an interesting task for future work could incorporate both of these modalities together in order to achieve a more comprehensive description of the song. This would involve transcribing the lyrics of the songs in the dataset, as well as extracting additional features which describe how the vocals are sung.

The utilization of dynamic features is also an interesting option to consider for future tasks. Instead of creating static statistical features from the dynamic data (mean, median, min, max, std, skew, kurtosis), the time series data itself could be used with a RNN model. This would allow the temporal changes in the music to affect emotion over time. This approach comes with its own limitations however; the dimensionality of the feature set would be much greater, and well-established feature selection algorithms for time series data like this is not readily available. Additionally, knowing which temporal aspects of the songs have meaning and which do not would be an important task. As such, the large complexity of this data without selection could contain a lot of noise and learning might be difficult.

In order to combat these issues, dimensionality reduction techniques like autoencoders would help by creating latent representations of the dynamic features. This latent space would be a smaller representation of the feature space, but would still include relevant information. Finally, adding attention to the encoder would help to identify which time frames are most important.

## 7.3 Conclusion

Throughout this study, we find through the use of various machine learning models, that our feature set outperforms the state-of-the-art on the 4Q dataset for music emotion recognition. Our best approach utilized a MLP model trained on a total of 300 acoustical features to predict the four quadrants of the valence and arousal graph, resulting in an $F_1$-score of 82.03%. This indicates that more than 100 features are necessary to discern emotion, and strong learning models are needed to classify these high dimensional spaces. As the field of MER progresses further, automatic emotion tagging will enable the labeling of large music libraries, something that is not feasible with current manual annotation techniques. This would allow for advancements in music recommendation systems, mood based playlist generation [18], affective music therapy [27], and generative music composition [16].

# 8 Appendix

The four tables below contain the features that are a part of our best top 100 feature set after feature selection, and Panda's et al. top 100 feature set after ReliefF feature selection. Features are considered present in Pandas if the feature type is the same, regardless of band number or statistic type. For example if MFCC 0 min was in Pandas feature set and MFCC 3 mean is in ours, then ours would be considered in Panda's.

| Feature Name | Musical Dimension | Toolkit | Present in Pandas |
|---|---|---|---|
| Pitch Salience skew | Melody | Essentia | No |
| Pitch Centroid Max | Melody | Essentia | No |
| Pure Tonalness max | Melody | PsySound3 | No |
| SWIPEP Pitch Strength kurtosis | Melody | PsySound3 | Yes |
| Inharmonicity kurtosis | Harmony | Essentia | Yes |
| Chords Progression max | Harmony | Essentia | No |
| Key Scale mean | Harmony | Essentia | No |
| Key Strength max | Harmony | Essentia | No |
| HPCP Crest kurtosis | Harmony | Essentia | No |
| Key $\text{Strength}_{1,3}$ skew | Harmony | MIR TB | No |
| Key $\text{Strength}_{1,9}$ skew | Harmony | MIR TB | No |
| Key $\text{Strength}_{2,5}$ skew | Harmony | MIR TB | No |
| Key $\text{Strength}_{2,8}$ skew | Harmony | MIR TB | No |
| Key $\text{Strength}_{2,12}$ skew | Harmony | MIR TB | No |
| Key min | Harmony | MIR TB | No |
| HCDF median | Harmony | MIR TB | No |
| Chord Change Likelihood std | Harmony | PsySound3 | Yes |
| Mean Peak Ratio Chroma F# kurtosis | Harmony | MARSYAS | No |
| Tonal $\text{Centroid}_4$ kurtosis | Harmony | MIR TB | Yes |
| Tonal $\text{Centroid}_6$ kurtosis | Harmony | MIR TB | Yes |
| Onset Rate mean | Rhythm | Essentia | No |
| $\text{Fluctuation}_{13}$ min | Rhythm | MIR TB | Yes |
| Tempo M mean | Rhythm | MIR TB | No |
| Pulse Clarity median | Rhythm | MIR TB | No |
| $\text{Loudness}_2$ skew | Dynamics | Yaafe | Yes |
| $\text{Loudness}_{10}$ min | Dynamics | Yaafe | Yes |
| $\text{Loudness}_{17}$ max | Dynamics | Yaafe | Yes |
| Perceptual $\text{Spread}_{10}$ skew | Dynamics | Yaafe | No |
| $\text{Barkbands}_{12}$ mean | Tone Color | Essentia | No |
| $\text{Barkbands}_{16}$ kurtosis | Tone Color | Essentia | No |
| $\text{Barkbands}_{17}$ skew | Tone Color | Essentia | No |
| $\text{MFCC}_5$ skew | Tone Color | Essentia | Yes |
| $\text{MFCC}_7$ max | Tone Color | Essentia | Yes |
| $\text{MFCC}_{11}$ skew | Tone Color | Essentia | Yes |
| $\text{MFCC}_6$ kurtosis | Tone Color | MIR TB | Yes |
| $\text{MFCC}_{11}$ min | Tone Color | MIR TB | Yes |
| $\text{MFCC}_{12}$ min | Tone Color | MIR TB | Yes |
| Spectral Energyband Low median | Tone Color | Essentia | No |
| $\text{Tristimulus}_2$ max | Tone Color | Essentia | No |
| Spectral $\text{Contrast}_1$ min | Tone Color | Essentia | No |

Table 20: Features from our top 100 feature set part 1.

| Feature Name | Musical Dimension | Toolkit | Present in Pandas |
|---|---|---|---|
| $GFCC_2$ skew | Tone Color | Essentia | No |
| $GFCC_3$ max | Tone Color | Essentia | No |
| $GFCC_3$ kurtosis | Tone Color | Essentia | No |
| $GFCC_6$ median | Tone Color | Essentia | No |
| $GFCC_8$ min | Tone Color | Essentia | No |
| $GFCC_9$ kurtosis | Tone Color | Essentia | No |
| $GFCC_{11}$ kurtosis | Tone Color | Essentia | No |
| Melband Crest max | Tone Color | Essentia | No |
| Power Cepstrum Mean kurtosis | Tone Color | Essentia | No |
| LPC $\text{Reflection}_2$ max | Tone Color | Essentia | No |
| LPC $\text{Reflection}_4$ max | Tone Color | Essentia | No |
| LPC $\text{Reflection}_5$ max | Tone Color | Essentia | No |
| LPC $\text{Reflection}_6$ min | Tone Color | Essentia | No |
| $LSP_2$ max | Tone Color | Essentia | Yes |
| $LSP_3$ kurtosis | Tone Color | Essentia | Yes |
| $LSP_6$ std | Tone Color | Essentia | Yes |
| $LSP_3$ mean | Tone Color | MARSYAS | Yes |
| $LSP_7$ std | Tone Color | MARSYAS | Yes |
| $LPCC_3$ max | Tone Color | MARSYAS | No |
| Roughness skew | Tone Color | MIR TB | No |
| Spectral Irregularity max | Tone Color | MIR TB | No |
| Spectral mean skew | Tone Color | PsySound3 | No |
| Tonal Dissonance HK std | Tone Color | PsySound3 | Yes |
| Tonal Dissonance S max | Tone Color | PsySound3 | Yes |
| $\text{Amplitude Modulation}_7$ skew | Tone Color | Yaafe | No |
| $SCF_7$ max | Tone Color | Yaafe | Yes |
| $SCF_{13}$ kurtosis | Tone Color | Yaafe | Yes |
| Mean $SCF_4$ skew | Tone Color | MARSYAS | Yes |
| Mean $SCF_8$ max | Tone Color | MARSYAS | Yes |
| Mean $SCF_9$ min | Tone Color | MARSYAS | Yes |
| Mean $SCF_9$ skew | Tone Color | MARSYAS | Yes |
| Std $SCF_5$ max | Tone Color | MARSYAS | Yes |
| Std $SCF_5$ kurtosis | Tone Color | MARSYAS | Yes |
| Std $SCF_8$ std | Tone Color | MARSYAS | Yes |
| Std $SCF_{12}$ kurtosis | Tone Color | MARSYAS | Yes |
| Std $SCF_{15}$ max | Tone Color | MARSYAS | Yes |
| $SFM_0$ mean | Tone Color | Yaafe | Yes |
| $SFM_1$ skew | Tone Color | Yaafe | Yes |
| $SFM_5$ skew | Tone Color | Yaafe | Yes |
| $SFM_5$ kurtosis | Tone Color | Yaafe | Yes |
| $SFM_{18}$ kurtosis | Tone Color | Yaafe | Yes |
| Mean $SFM_7$ max | Tone Color | MARSYAS | Yes |
| Mean $SFM_9$ max | Tone Color | MARSYAS | Yes |
| Mean $SFM_{13}$ skew | Tone Color | MARSYAS | Yes |
| Std $SFM_6$ std | Tone Color | MARSYAS | Yes |
| $\text{Zerocrossingrate}_0$ max | Tone Color | MARSYAS | No |
| EXPRESSIVE_ TECHNIQUES-Tremolo Notes in Cents (Max) | Expressivity | [49] | Yes |
| EXPRESSIVE_ TECHNIQUES-Vibrato to Non Vibrato Notes Ratio | Expressivity | [49] | Yes |
| EXPRESSIVE_TECHNIQUES-Vibrato Rate (Std) | Expressivity | [49] | Yes |
| EXPRESSIVE_TECHNIQUES-Vibrato Extent (Kurtosis) | Expressivity | [49] | Yes |
| EXPRESSIVE_TECHNIQUES-Glissando Direction (Up) | Expressivity | [49] | Yes |
| EXPRESSIVE_TECHNIQUES-ART-Other Notes Duration (Mean) | Expressivity | [49] | Yes |
| EXPRESSIVE_TECHNIQUES-Vibrato Higher Notes Coverage (C4+) | Expressivity | [49] | Yes |
| EXPRESSIVE_TECHNIQUES-Glissando Slope (Std) | Expressivity | [49] | Yes |
| ORIGINAL-TEXTURE-Musical Layers (Mean) | Texture | [49] | Yes |
| VOICE-TEXTURE-State Transitions ML0 - ML1 (Per Sec) | Texture | [49] | Yes |
| ORIGINAL-VAT-Probability of Creaky Voice (Skewness) | Voice | [49] | Yes |

Table 21: features from the our top 100 feature set part 2.

| Feature Name | Musical Dimension | Toolkit |
|---|---|---|
| MFCC 1 Mean | Tone Color | Marsyas |
| Average Power Spectrum (Median) | Tone Color | PsySound3 |
| ORIGINALTEXTURE-Musical Layers (Mean) | Texture | [49] |
| Spectral 2nd Moment (Median) | Tone Color | PsySound3 |
| Spectral Skewness (std) | Tone Color | MIR TB |
| Spectral Skewness (Max) | Tone Color | MIR TB |
| Rolloff (Mean) | Tone Color | Marsyas |
| ORIGINAL-TEXTURE-Musical Layers (Std) | Texture | [49] |
| Rolloff (MeanA/StdM) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Tremolo Notes in Cents (Mean) | Expressivity | [49] |
| SFM 15 (Mean) | Tone Color | Marsyas |
| Average Power Spectrum (Mean) | Tone Color | PsySound3 |
| Spectral Dissonance S (Skewness) | Tone Color | PsySound3 |
| MFCC 1 (Max) | Tone Color | Marsyas |
| ORIGINAL-TEXTURE-State Transitions ML1 - ML0 (Per Sec) | Texture | [49] |
| Spectral Entropy (Std) | Tone Color | MIR TB |
| FFT Spectrum Skewness (Median) | Tone Color | PsySound3 |
| Spectral Centroid (Std) | Tone Color | MIR TB |
| SFM 12 (Mean) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Vibrato Extent (Std) | Expressivity | [49] |
| EXPRESSIVE_TECHNIQUES-Tremolo Higher Notes Coverage (C4+) | Expressivity | [49] |
| MFCC 1 (StdA/MeanM) | Tone Color | Marsyas |
| Spectral Dissonance HK (Std) | Tone Color | PsySound3 |
| Inharmonicity (Std) | Harmony | MIR TB |
| Events Density | Rhythm | MIR TB |
| Inharmonicity (Mean) | Harmony | MIR TB |
| SCF 15 (Mean) | Tone Color | Marsyas |
| MFCC 1 (StdA/StdM) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Tremolo Notes in Cents (Max) | Expressivity | [49] |
| EXPRESSIVE_TECHNIQUES-Vibrato Rate (Kurtosis) | Expressivity | [49] |
| LSP 4 (Std) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Vibrato Base Freq (Kurtosis) | Expressivity | [49] |
| EXPRESSIVE_TECHNIQUES-Vibrato to Non Vibrato Notes Ratio | Expressivity | [49] |
| SFM 9 (Mean) | Tone Color | Marsyas |
| SCF 12 (Mean) | Tone Color | Marsyas |
| Rolloff (StdA/StdM) | Tone Color | Marsyas |
| SCF 13 (Mean) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Vibrato Base Freq (Min) | Expressivity | [49] |
| Fluctuation (Std) | Rhythm | MIR TB |
| EXPRESSIVE_TECHNIQUES-Vibrato Length (Kurtosis) | Expressivity | [49] |
| MFCC 0 (StdA/MeanM) | Tone Color | Marsyas |
| Fluctuation (Max) | Rhythm | MIR TB |
| MFCC 1 (Std) | Tone Color | Marsyas |
| VOICE-TEXTURE-State Transitions ML0 - ML1 (Per Sec) | Texture | [49] |
| EXPRESSIVE_TECHNIQUES-Vibrato Rate (Std) | Expressivity | [49] |
| LSP 5 (Std) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Vibrato Base Freq (Std) | Expressivity | [49] |
| Brightness (Std) | Tone Color | MIR TB |
| SWIPEP Pitch Strength (Mean) | Melody | PsySound3 |
| EXPRESSIVE_TECHNIQUES-Vibrato Extent (Kurtosis) | Expressivity | [49] |
| TEXTURE-Musical Layers (Max) | Texture | [49] |
| SCF 11 (Mean) | Tone Color | Marsyas |
| TEXTURE-ML1-Monophonic Texture (Percentage) | Texture | [49] |
| SFM 15 (Std) | Tone Color | Marsyas |
| Tonal Dissonance HK (Std) | Tone Color | PsySound3 |
| Brightness (Min) | Tone Color | MIR TB |
| LSP 1 (Mean) | Tone Color | Marsyas |
| SCF 9 (Mean) | Tone Color | Marsyas |
| MFCC 1 (MeanA/StdM) | Tone Color | Marsyas |
| SCF 10 (Mean) | Tone Color | Marsyas |
| Brightness (Skewness) | Tone Color | MIR TB |
| TEXTURE-State Transitions ML2 - ML3 (Per Sec) | Texture | [49] |
| LSP 3 (Min) | Tone Color | Marsyas |
| Loudness (Skewness) | Dynamics | PsySound3 |

Table 22: Panda's top 100 feature subset after ReliefF feature selection and Novel MIDI features added part 1.

| Feature Name | Musical Dimension | Toolkit |
|---|---|---|
| SFM 17 (Std) | Tone Color | Marsyas |
| SFM 9 (Skewness) | Tone Color | Marsyas |
| MFCC 0 (MeanA/StdM) | Tone Color | Marsyas |
| SFM 10 (Skewness) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Vibrato Length (Skewness) | Expressivity | [49] |
| EXPRESSIVE_TECHNIQUES-ART-Other Notes Duration (Mean) | Expressivity | [49] |
| Spectral Entropy (Skewness) | Tone Color | MIR TB |
| SFM 6 (Mean) | Tone Color | Marsyas |
| TEXTURE-State Transitions ML1 - ML2 (Per Sec) | Texture | [49] |
| Terhardt Chord Change Likelyhood (Median) | Melody | PsySound3 |
| EXPRESSIVE_TECHNIQUES-Glissando Extent (Std) | Expressivity | [49] |
| Spectral Flux (Skewness) | Tone Color | MIR TB |
| Tonal Dissonance S (Skewness) | Tone Color | PsySound3 |
| SCF 16 (Std) | Tone Color | Marsyas |
| LSP 4 (Min) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Tremolo Notes in Cents (Min) | Expressivity | [49] |
| Sharpness (Std) | Harmony | PsySound3 |
| SCF 15 (Std) | Tone Color | Marsyas |
| Spectral Dissonance HK (Max) | Tone Color | PsySound3 |
| Volume (Skewness) | Dynamics | PsySound3 |
| EXPRESSIVE_TECHNIQUES-Vibrato Higher Notes Coverage (C4+) | Expressivity | [49] |
| SharpnessA (Skewness) | Harmony | PsySound3 |
| VAT-Probability of Creaky Voice (Skewness) | Voice | [49] |
| SFM 7 (Mean) | Tone Color | Marsyas |
| Inharmonicity (Min) | Harmony | MIR TB |
| Short-Term Loudness (Skewness) | Dynamics | PsySound3 |
| Fluctuation (Skewness) | Rhythm | MIR TB |
| SFM 8 (Skewness) | Tone Color | Marsyas |
| SFM 7 (Skewness) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Glissando Length (Std) | Expressivity | [49] |
| Tonal Centroid 3 (Std) | Harmony | MIR TB |
| SCF 17 (Mean) | Tone Color | Marsyas |
| LSP 5 (Min) | Tone Color | Marsyas |
| SFM 11 (Skewnss) | Tone Color | Marsyas |
| EXPRESSIVE_TECHNIQUES-Glissando Slope (Std) | Expressivity | [49] |

Table 23: Panda's top 100 feature subset after ReliefF feature selection and Novel MIDI features added part 2.

# References


[1] Harris Drucker et al. “Support Vector Regression Machines”. In: *Proceedings of the 9th International Conference on Neural Information Processing Systems (NeurIPS)* (1996), pp. 155–161.

[2] Anna Aljanaki, Yi-Hsuan Yang, and Mohammad Soleymani. “Developing a benchmark for emotional analysis of music”. In: *Plos One* (2017). DOI: 10.1371/journal.pone.0173392.

[3] W. Aures. “Berechnungsverfahren für den sensorischen Wohlklang beliebiger Schallsignale”. In: *Acustica 59* (1985), pp. 130–141.

[4] Mathieu B. et al. “YAAFE, an Easy to Use and Effcient Audio Feature Extraction Software”. In: *11th International Society for Music Information Retrieval Confrence (ISMIR)*. 2010. DOI: 10.5281/zenodo.1418320.

[5] Laura-Lee Balkwill and William Thompson. “A Cross-Cultural Investigation of the Perception of Emotion in Music: Psychophysical and Cultural Cues”. In: *Music Perception* 17 (Oct. 1999), pp. 43–64. DOI: 10.2307/40285811.

[6] Mario Beraha et al. “Feature Selection via Mutual Information: New Theoretical Insights”. In: *International Joint Conference on Neural Networks (IJCNN)*. July 2019, pp. 1–9. DOI: 10.1109/IJCNN.2019.8852410.

[7] Dmitry Bogdanov et al. “ESSENTIA: an open-source library for sound and music analysis”. In: *Proceedings of the 21st ACM International Conference on Multimedia (ACM-MM)*. MM ’13. Barcelona, Spain: Association for Computing Machinery, 2013, pp. 855–858. ISBN: 9781450324045. DOI: 10.1145/2502081.2502229.

[8] Paul M. Brossier. “Automatic annotation of musical audio for interactive applications”. PhD thesis. Queen Mary University of London, 2007.

[9] Densil Cabrera et al. “PsySound3: a program for the analysis of sound recordings”. In: *The Journal of the Acoustical Society of America (JASA)* 123 (June 2008), p. 3247. DOI: 10.1121/1.2933513.

[10] Arturo Camacho and John Harris. “A sawtooth waveform inspired pitch estimator for speech and music”. In: *The Journal of the Acoustical Society of America (JASA)* 124 (Oct. 2008), pp. 1638–52. DOI: 10.1121/1.2951592.

[11] Josef Chalupper and Hugo Fastl. “Dynamic Loudness Model (DLM) for Normal and Hearing-Impaired Listeners”. In: *Acta Acustica united with Acustica* 88 (May 2002), pp. 378–386.

[12] George Soci Charles Osgood and Percy TannenBaum. “The dimensionality of the semantic space”. In: *The Measurement of Meaning* (1957), pp. 1–30.

[13] Yu-An Chen et al. “The AMG1608 dataset for music emotion recognition”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 693–697. DOI: 10.1109/ICASSP.2015.7178058.

[14] Shreyan Chowdhury et al. “Towards Explainable Music Emotion Recognition: The Route via Mid-level Features”. In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2019. arXiv: 1907.03572 [cs.SD].

[15] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine Learning* 20 (1995), pp. 273–297. DOI: 10.1007/BF00994018.

[16] Adyasha Dash and Kathleen Agres. “AI-Based Affective Music Generation Systems: A Review of Methods and Challenges”. In: *ACM Comput. Surv.* 56.11 (July 2024), pp. 1–34. ISSN: 0360-0300. DOI: 10.1145/3672554.

[17] S. Davis and P. Mermelstein. “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.4 (1980), pp. 357–366. DOI: 10.1109/TASSP.1980.1163420.

[18] Rémi Delbouys et al. “Music Mood Detection Based On Audio And Lyrics With Deep Neural Net”. In: *International Conference on Music Information Retrieval (ISMIR)*. Sept. 2018. DOI: 10.5281/zenodo.1492426.

[19] Yizhuo Dong et al. “Bidirectional Convolutional Recurrent Sparse Network (BCRSN): An Efficient Model for Music Emotion Recognition”. In: *IEEE Transactions on Multimedia* 21.12 (2019), pp. 3150–3163. DOI: 10.1109/TMM.2019.2918739.

[20] Karin Dressler. “Automatic transcription of the melody from polyphonic music”. en. PhD thesis. Ilmenau: Technical University of Ilmenau, July 2017.

[21] Paul Ekman. “An Argument for Basic Emotions”. In: *Cognition and Emotion* (1992), pp. 169–200.

[22] Antti J. Eronen. “Automatic musical instrument recognition”. MA thesis. Tampere University of Technology, 2001.

[23] Slim Essid. “Classification automatique des signaux audio-fréquences : reconnaissance des instruments de musique”. Theses. Université Pierre et Marie Curie - Paris VI, Dec. 2005.

[24] Yazhong Feng, Yueting Zhuang, and Yunhe Pan. “Music information retrieval by detecting mood via computational media aesthetics”. In: *Proceedings IEEE/WIC International Conference on Web Intelligence (WI)*. IEEE, 2003, pp. 235–241. DOI: 10.1109/WI.2003.1241199.

[25] Alicia Fernández-Sotos, Antonio Fernández-Caballero, and José Miguel Latorre. “Influence of Tempo and Rhythmic Unit in Musical Emotion Regulation”. In: *Frontiers in Computational Neuroscience* 10 (2016). DOI: 10.3389/fncom.2016.00080.

[26] Aalf Gabrielsson and Erik Lindström. "The Role of Structure in the Musical Expression of Emotions". In: *Handbook of music and emotion: Theory, research, applications* (Jan. 2010), pp. 367–400. DOI: 10.1093/acprof:oso/9780199230143.003.0014.

[27] Shaurya Gaur and Patrick Donnelly. "Mood Dynamic Playlist: Interpolating a musical path between emotions using a KNN algorithm". In: *Iternational Conference on Human Iteraction & Emerging Technologies: Artificial Intelligence*. 2022.

[28] Isabelle Guyon et al. "Gene Selection for Cancer Classification using Support Vector Machines". In: *Machine Learning* 46 (2002), pp. 389–422. DOI: 10.1023/A:1012487302797.

[29] Christopher Harte, Mark Sandler, and Martin Gasser. "Detecting harmonic change in musical audio". In: *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM)*. AMCMM '06. Santa Barbara, California, USA: Association for Computing Machinery, 2006, pp. 21–26. ISBN: 1595935010. DOI: 10.1145/1178723.1178727.

[30] Tin Kam Ho. "Random decision forests". In: *Proceedings of 3rd International Conference on Document Analysis and Recognition (ICDAR)*. Vol. 1. 1995, 278–282 vol.1. DOI: 10.1109/ICDAR.1995.598994.

[31] Patrik N. Juslin. "Perceived Emotional Expression in Synthesized Performances of a Short Melody: Capturing the Listener's Judgment Policy". In: *Musicae Scientiae* 1.2 (1997), pp. 225–256. DOI: 10.1177/102986499700100205.

[32] Min-Jae Kim et al. "Functionality-Based Processing-In-Memory Accelerator for Deep Convolutional Neural Networks". In: *IEEE Access* (Oct. 2021), pp. 1–1. DOI: 10.1109/ACCESS.2021.3122818.

[33] Igor Kononenko. "Estimating Attributes: Analysis and Extensions of RELIEF". In: *Machine Learning (ECML)*. Vol. 784. Lecture Notes in Computer Science. 1994, pp. 171–182.

[34] Olivier Lartillot and Didier Grandjean. "Tempo and Metrical Analysis by Tracking Multiple Metrical Levels Using Autocorrelation". In: *Applied Sciences* 9.23 (2019). ISSN: 2076-3417. DOI: 10.3390/app9235121.

[35] Olivier Lartillot, Petri Toiviainen, and Tuomas Eerola. "A Matlab Toolbox for Music Information Retrieval". In: *Data Analysis, Machine Learning and Applications*. Ed. by Christine Preisach et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 261–268. ISBN: 978-3-540-78246-9.

[36] Olivier Lartillot et al. "A Simple, High-Yield Method for Assessing Structural Novelity". In: *Proceedings of the 3rd International Conference on Music and Emotion (ICME3)*. Jyväskylä, Finland, 2013.

[37] Olivier Lartillot et al. “Multi-Feature Modeling of Pulse Clarity: Design, Validation and Optimization”. In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2008, pp. 521–526. DOI: 10.5281/zenodo.1415513.

[38] Edith Law et al. “Evaluation of Algorithms Using Games: The Case of Music Tagging.” In: *Proceedings of the 10th International Society for Music Information Retrieval (ISMIR)*. Kobe, Japan, 2009, pp. 387–392. DOI: 10.5281/zenodo.1417646.

[39] Xinxing Li et al. “A deep bidirectional long short-term memory based multi-scale approach for music dynamic emotion prediction”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 544–548. DOI: 10.1109/ICASSP.2016.7471734.

[40] Xinxing Li et al. “DBLSTM-based multi-scale fusion for dynamic emotion prediction in music”. In: *IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2016, pp. 1–6. DOI: 10.1109/ICME.2016.7552956.

[41] M. S. Likitha et al. “Speech based human emotion recognition using MFCC”. In: *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. 2017, pp. 2257–2260. DOI: 10.1109/WiSPNET.2017.8300161.

[42] Lie Lu, D. Liu, and Hong-Jiang Zhang. “Automatic mood detection and tracking of music audio signals”. In: *IEEE Transactions on Audio, Speech, and Language Processing (TASLPRO)* 14.1 (2006), pp. 5–18. DOI: 10.1109/TSA.2005.860344.

[43] Miroslav Malík et al. “Stacked Convolutional and Recurrent Neural Networks for Music Emotion Recognition”. In: *Sound and Music Computing Conference (SMC)*. Vol. 14. Espoo, Finland, July 2017. DOI: 10.48550/arXiv.1706.02292.

[44] Brian C. J. Moore and Brian R. Glasberg. “Suggested formulae for calculating auditory-filter bandwidths and excitation patterns”. In: *The Journal of the Acoustical Society of America* 74.3 (Sept. 1983), pp. 750–753. ISSN: 0001-4966. DOI: 10.1121/1.389861.

[45] Brian C. J. Moore, Brian R. Glasberg, and Thomas Baer. “A model for the prediction of thresholds, loudness, and partial loudness”. In: *Journal of The Audio Engineering Society (AES)* 45 (1997), pp. 224–240.

[46] Andrew Y. Ng. “Feature selection, L1 vs. L2 regularization, and rotational invariance”. In: *Proceedings of the 21st International Conference on Machine Learning (ICML)*. ICML 2004. Banff, Alberta, Canada: Association for Computing Machinery, 2004, p. 78. ISBN: 1581138385. DOI: 10.1145/1015330.1015435.

[47] Rui Pedro Paiva, Teresa Mendes, and Amílcar Cardoso. "Melody Detection in Polyphonic Musical Signals: Exploiting Perceptual Rules, Note Salience, and Melodic Smoothness". In: *Computer Music Journal* 30.4 (Dec. 2006), pp. 80–98. ISSN: 0148-9267. DOI: 10.1162/comj.2006.30.4.80.

[48] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. "Audio Features for Music Emotion Recognition: A Survey". In: *IEEE Transactions on Affective Computing* 14.1 (2023), pp. 68–88. DOI: 10.1109/TAFFC.2020.3032373.

[49] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. "Novel Audio Features for Music Emotion Recognition". In: *IEEE Transactions on Affective Computing* 11.4 (2020), pp. 614–626. DOI: 10.1109/TAFFC.2018.2820691.

[50] Renato Panda and Rui Pedro Paiva. "MIREX 2012: Mood Classification Tasks Submission". In: *Music Information Retrieval Exchange (MIREX)*. Porto, Portugal, Oct. 2012.

[51] Richard Parncutt. *Harmony: A Psychoacoustical Approach*. Springer Berlin, Heidelberg, 1989.

[52] Geoffroy Peeters et al. "A large set of audio features for sound description (similarity and classification) in the CUIDADO project". In: *1st Project Report ICRAM* 54.0 (2004), pp. 1–25.

[53] Robert Plutchik. "A General Psychoevolutionary Theory of Emotion". In: *Theories of Emotion* (1980). Ed. by Henry Kellerman, pp. 3–33.

[54] Jordi Pons, Thomas Lidy, and Xavier Serra. "Experimenting with musically motivated convolutional neural networks". In: *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*. 2016, pp. 1–6. DOI: 10.1109/CBMI.2016.7500246.

[55] James Russell. "A Circumplex Model of Affect". In: *Journal of Personality and Social Psychology* 39 (Dec. 1980), pp. 1161–1178. DOI: 10.1037/h0077714.

[56] Justin Salamon and Emilia Gomez. "Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.6 (2012), pp. 1759–1770. DOI: 10.1109/TASL.2012.2188515.

[57] Ricardo Scholz, Geber Ramalho, and Giordano Cabral. "Cross Task Study on MIREX Recent Results: an Index for Evolution Measurement and Some Stagnation Hypotheses". In: *17th International Society for Music Information Retrieval Conference (ISMIR)*. Aug. 2016. DOI: 10.5281/zenodo.1416897.

[58] M. Schuster and K.K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681. DOI: 10.1109/78.650093.

[59] R Steuer et al. “The Mutual Information: Detecting and Evaluating Dependencies between Variables”. In: *Bioinformatics* 18 Suppl 2 (Feb. 2002), S231–40. DOI: 10.1093/bioinformatics/18.suppl_2.S231.

[60] S. S. Stevens. “Neural Events and the Psychophysical Law”. In: *Science* 170.3962 (1970), pp. 1043–1050. DOI: 10.1126/science.170.3962.1043.

[61] S. S. Stevens, J. Volkmann, and E. B. Newman. “A Scale for the Measurement of the Psychological Magnitude Pitch”. In: *The Journal of the Acoustical Society of America* 8.3 (Jan. 1937), pp. 185–190. ISSN: 0001-4966. DOI: 10.1121/1.1915893.

[62] Sebastian Streich and Perfecto Herrera. “Detrended Fluctuation Analysis of Music Signals: Danceability Estimation and further Semantic Characterization”. In: *Proceedings of the 118th AES Convention*. Barcelona, Spain, May 2005.

[63] Ernst Terhardt, Gerhard Stoll, and Manfred Seewann. “Algorithm for extraction of pitch and pitch salience from complex tonal signals”. In: *The Journal of the Acoustical Society of America (JASA)* 71.3 (Mar. 1982), pp. 679–688. ISSN: 0001-4966. DOI: 10.1121/1.387544.

[64] George Tzanetakis and Perry Cook. “MARSYAS: a framework for audio analysis”. In: *Organised Sound* 4 (Sept. 2002), pp. 169–175. DOI: 10.1017/S1355771800003071.

[65] Ryan J. Urbanowicz et al. “Relief-based feature selection: Introduction and review”. In: *Journal of Biomedical Informatics* 85 (2018), pp. 189–203. DOI: 10.1016/j.jbi.2018.07.014.

[66] Amy Warriner, Victor Kuperman, and Marc Brysbaert. “Norms of valence, arousal, and dominance for 13,915 English lemmas”. In: *Behavior research methods* 45 (Feb. 2013), pp. 1191–1207. DOI: 10.3758/s13428-012-0314-x.

[67] Gregory Webster and Catherine Weir. “Emotional Responses to Music: Interactive Effects of Mode, Texture, and Tempo”. In: *Motivation and Emotion* 29 (Mar. 2005), pp. 19–39. DOI: 10.1007/s11031-005-4414-0.

[68] Bin Wu, Andrew Horner, and Chung Lee. “The Correspondence of Music Emotion and Timbre in Sustained Musical Instrument Sounds”. In: *Journal of the Audio Engineering Society (AES)* 62 (Oct. 2014), pp. 663–675. DOI: 10.17743/jaes.2014.0037.

[69] Tien-Lin Wu and Shyh-Kang Jeng. “Probabilistic Estimation of a Novel Music Emotion Model”. In: *Advances in Multimedia Modeling*. Ed. by Shin’ichi Satoh, Frank Nack, and Minoru Etoh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 487–497. ISBN: 978-3-540-77409-9.

[70] Dan Yang and Won-Sook Lee. “Disambiguating Music Emotion Using Software Agents.” In: *5th International Conference on Music Information Retrieval (ISMIR)*. Barcelona, Spain, 2004. DOI: 10.5281/zenodo.1415271.

[71] Yi-Hsuan Yang et al. “A Regression Approach to Music Emotion Recognition”. In: *IEEE Transactions on Audio, Speech, and Language Processing (TASLPRO)* 16.2 (2008), pp. 448–457. DOI: 10.1109/TASL.2007.911513.

[72] Yong Yu et al. “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures”. In: *Neural Computation* 31.7 (2019), pp. 1235–1270. DOI: 10.1162/neco_a_01199.

[73] Matthew D. Zeiler and Rob Fergus. “Visualizing and Understanding Convolutional Networks”. In: *Computer Vision (ECCV)*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 818–833.

[74] Kejun Zhang et al. “The PMEmo Dataset for Music Emotion Recognition”. In: *International Conference on Multimedia Retrieval (ACM MIR)*. ICMR ’18. Yokohama, Japan: ACM, 2018, pp. 135–142. ISBN: 978-1-4503-5046-4. DOI: 10.1145/3206025.3206037.

[75] Eberhard Zwicker and Hugo Fastl. *Psychoacoustics: Facts and Models*. Springer Berlin, Heidelberg, 1999.