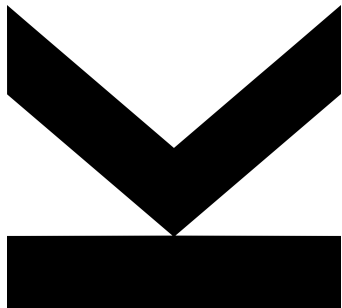


Music Emotion Recognition: Uni- and Multi-modal Machine Learning Approaches based on Audio and Symbolic Data



Bachelor Thesis

to obtain the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

Author

Nina Braunmiller

Matriculation number

k11923286

Submission

**Institute of
Computational
Perception**

Thesis Supervisor / First
Supervisor

**Prof. Dr. Emilia
Parada-Cabaleiro**

Second Supervisor

**Univ.-Prof. Mag. Dr.
Markus Schedl**

September 2023

**JOHANNES KEPLER
UNIVERSITY LINZ**

Altenbergerstraße 69

4040 Linz, Austria

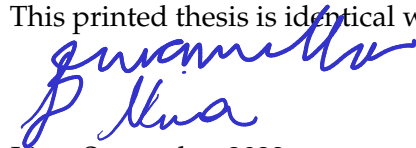
www.jku.at

DVR 0093696

Statutory Declaration

I hereby declare that the thesis submitted is my own unaided work, that I have not used other than the sources indicated, and that all direct and indirect sources are acknowledged as references.

This printed thesis is identical with the electronic version submitted.

A handwritten signature in blue ink, appearing to read 'Nina Braunmiller', is written over the printed name.

Linz, September 2023

Abstract

MER research attempts to measure and predict the emotions of music listeners. The underlying work uses the GEMS annotated EMMA database. It extracts established audio features from the provided tracks. It also adds symbolic data to the database by collecting and generating MIDI files so that symbolic features can also be extracted. In the next step, the uni-modal models kNN, RF and SVM, as well as a custom multi-modal architecture, are trained to predict the GEMS labels in a supervised setting. It turns out that while there is some success in prediction, there are limitations, such as the trade-off between the quality and quantity of data, but also the computational complexity, which prevent finding optimal solutions.

Contents

1	Introduction	1
2	Related works	3
2.1	Underlying emotion theory	3
2.2	Features of interest in MER	5
2.2.1	Audio features	6
2.2.2	Symbolic features	7
2.2.3	Further features	8
2.3	Common AI models in MER	8
2.3.1	Models concerning scalar features	8
2.3.2	CNN based models	8
2.3.3	Multi-modal models	10
3	Method	12
3.1	Dataset principles	12
3.1.1	Database characteristics	12
3.1.2	Label categorization	13
3.2	MIDI data collection and generation	16
3.2.1	Web scraping of MIDI files	16
3.2.2	audio2MIDI converter	18
3.3	Feature extraction for symbolic data	20
3.3.1	Manual feature extraction inspired by Panda, Malheiro, and Paiva [20]	20
3.3.2	jSymbolic	22
3.4	Feature extraction for audio data	23
3.4.1	Statistical features	23
3.4.2	Functional features	23
3.4.3	LLD features	24
3.5	Models	24
3.5.1	Uni-modal models	24
3.5.2	Multi-modal model architectures	25
4	Results	27
4.1	The top 30 scalar features	27

4.2	Performance of the uni-modal model	27
4.3	Performance of the multi-modal model	31
5	Discussion	33
5.1	Overview with limitations	33
5.2	Limitations in the MER research field	34
5.3	Conclusions and future work	35

List of Figures

2.1	GEMS: First-order factors (dimensions) (left) and belonging second-order factors (right) [38, p. 507].	5
3.1	Samples as their categorized label combination [sublimity, vitality, unease] which are determined with the help of the median with a threshold percentile of 65.	15
3.2	Binned value distribution for (a) sublimity, (b) vitality, and (c) unease. . .	15
3.3	Boxplots for (a) sublimity, (b) vitality, and (c) unease.	15
3.4	Visualization of the used multi-modal model which starts with the inputs (left) and ends up with the outputs (right). It depends on the variables in the legend box how the architecture looks in detail.	26

List of Tables

4.1	Top 30 k using the tools RFE and SelectFromModel of the package sklearn.feature_selection where each tool searches independently the top 20 features based on the regressive labeled MIDI files containing audio and symbolic scalar features.	28
4.2	The results of the best performing uni-modal estimators on the top 30 features considering the different data sources.	29
4.3	Summary of the used hyper-parameter settings whose estimators' results are collected in table 4.2. kNN: number of neighbors and weights; SVM: C, gamma, and kernel; RF: number estimators and criterion.	30
4.4	Best performing multi-modal models on the top 30 scalar features concerning the test set based on the data sources of the automatically generated MIDI files [34] and the music tracks.	31

1 Introduction

The power of music to enhance mood is well-known [26] and it can be found in various media contexts such as advertisements, movies, news channels, songs, and podcasts. This knowledge can be utilized to fulfill the emotional needs of customers. However, the challenge is to select appropriate music. Therefore, a tool that predicts the emotions induced in listeners can be crucial for the success of some companies. The try to recognize emotional states based on music with the help of signal processing and machine learning is also known as Music Emotion Recognition (MER). It is a sub-topic of the research field of Music Information Retrieval (MIR) [8, 10]. Different disciplines, like machine learning, music psychology, music theory, neuroscience, and signal processing, are engaged to come up with the best emotion estimation results [7]. The process of Music Emotion Recognition (MER) can be divided into three stages. The first stage involves defining the domain and selecting the appropriate dataset and emotion model. The dataset is used to predict emotions, while the emotion model specifies the types of emotions that should be predicted. In the second stage, features are extracted from the dataset, which play a crucial role in predicting emotions. The final stage is the emotion recognition stage, where a neural network is trained and executed to produce the final predictions.

With having a look at the feature and annotation choice, the underlying work brings in new ideas. When working with music in an automatic way, digital music representation is crucial. It's quite common to rely on information that is extracted from audio files [e.g. 3, 10, 22, 35, 36, 39]. However, these works miss the fact that a rule-based symbolic music representation may be more accurate with less noise potential. One storage format is the "Musical Instrument Digital Interface" [19, p. 1] (MIDI). For three decades the MIDI format has been widely used in the internet and science [19, 30]. The underlying thesis considers both, the audio files and the MIDI, as input sources.

Even further, the state-of-the-art annotation emotion model is the so-called Russell's circumplex model of affect which describes all human emotional states in a two-dimensional

space [25]. However, the underlying thesis breaks with that standard as it allows a more innovative emotion model, namely the "Geneva Emotional Music Scale" [38, p. 506] (GEMS). This shall allow us to measure the music-induced emotions instead of the perceived emotions as is the case for Russell's model [8].

To gain a better understanding of how music affects us, it is important to accurately measure music-induced emotions. Chapter 2 details various theories and approaches in the Music Emotion Recognition (MER) research field, including emotion theory for label design, feature choice, and Artificial Intelligence (AI) approaches. The following chapter, 3, presents the implementations of this thesis, including data generation, symbolic and audio feature extractions, and model setup. The results are listed in 4. Lastly, the limitations and future outlook of MER research are discussed.

2 Related works

The stages of MER research provide the fundamental steps to create a sufficient paper [8]. From this follows the structure of the current chapter. First, the underlying emotion model needs to be discussed. Afterward, the focus is concentrated on features. Last, AI models are introduced to give an overview of the research field and mark the ideas that are also implemented by the underlying thesis.

2.1 Underlying emotion theory

The emotion model is essential and should rely on scientific principles rather than arbitrary choices. It specifies how the annotated labels should be designed. A very well-known, state-of-the-art work is Russell's circumplex model of affect [25, 36]. The author considers lots of emotion theoretic studies in his assumptions. Due to the progress in this field, he can develop a two-dimensional, bi-polar, continuous, coordinate system-like emotion model. Its x-axis relates to valence which can be described as pleasure and the y-axis represents arousal. The underlying assumption is that different emotions are correlated with each other. It shall be possible to place them into the above-described coordinate-like system where they are evaluated due to their pleasure and arousal scores [25]. In several studies Russell [25] tries out several scaling methods. Students are asked to sort 28 emotion words into eight categories which represent a centered ball-like structure in the aforementioned emotion space. The more similar emotions are, the closer they are located to each other. Contrary affective states face each other. This shall be theoretically possible with all emotions [25]. Russell [25] goes on with further studies. The resulting models are quite similar. Different statistical approaches come up with similar solutions. Also, the last study supports the findings by using a principal component analysis. The biggest two components are valence and arousal. Further, less important axes can be found: sleepiness,

anger, and fear. However, the first two components valence and arousal already explain 45.8 % of the model variance [25].

This emotion model is widely used in MER as it is originally stated in the dimensional form [e.g. 5, 6, 26, 33, 36, 39]. Further going, the four quadrants of the dimensional coordinate system-like space can be defined as four emotion groups which can be predicted in a binary setting [e.g. 10, 26, 35].

However, the model has also its weak points. It is developed in a non-musical context [25]. Even further, it only considers perceived emotions but this thesis is interested in music-induced emotions [8]. Perceived emotions are the transported emotions through musical characteristics [7]. In contrast, induced emotions trigger emotions in listeners [7].

As induced emotions seem to be better suited to fulfill customers' needs, the underlying work selects the music-induced emotion model "Geneva Emotional Music Scale" [38, p. 506] (GEMS) [8]. Zentner, Grandjean, and Scherer [38] focus in detail on the topic of how to define a music-induced emotion model where emotion is simply defined as feeling. The model is built up by close investigation in the form of four studies [38].

In the first trial, a list consisting of affectional expressions and words regarding emotion in the context of music is created. Students rate the words to familiarity and whether they would prefer the current feeling compared to other ones. If both criteria are fulfilled for the majority of students then the words are kept. This results in a 146-item long list [38]. The second investigation discovers whether the final list of the first study is music-relevant. For this purpose, emotional ratings are collected. Those are centered around three separate situations: the perceived emotions by favorite music genre, the induced emotions by favorite music genre, and the experienced emotions in daily life. Only the emotions that are perceived more often than "never" in the mean are kept. 10 first-order factors are formulated. Each covers several items. Those turn out to be 10 GEMS dimensions [38]. The third study makes sure that the correlation among the factors stays low. According to statistical techniques, the model can be described by the fig. 2.1. From the dimensions follow the factors sublimity, vitality, and unease which are used as labels of the underlying thesis.

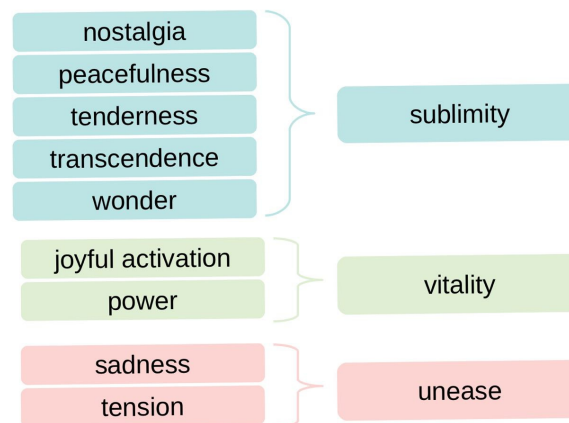


Figure 2.1: GEMS: First-order factors (dimensions) (left) and belonging second-order factors (right) [38, p. 507].

This section shows that the validity of the object to measure is essential. Otherwise, the prediction accuracy might suffer. Beyond the label quality, the selected features for the MER task are central. The upcoming section discusses the different usable feature types.

It is striking that a different feature choice can lead to a big gap in prediction performance [3, 22, 40]. For instance, a F1-Score enhancement of 9 % is possible [20].

2.2 Features of interest in MER

It depends on the data type which features are extractable. Also for the features, it is important to have some underlying theory of how music can be split into different feature types to adequately represent them. Panda, Malheiro, and Paiva [20] fix eight groups of music-relevant features: dynamics, expressive techniques, harmony, melody, musical form, musical texture, rhythm, and tone color. The features which belong to the tone color are over-represented in number whereas expressivity, texture, and form are under-represented [21]. It is evident that various papers figure out different feature categories as the most relevant ones. For instance, Panda, Malheiro, and Paiva [20] outline rhythm, texture, tone color, and expressive features as essential, whereas Panda, Malheiro, and Paiva [21] claim that harmonic, melodic, and rhythmic features are the most important ones in MER. Even more extreme, Russo et al. [26] state that no feature carries the most information.

The abstract feature categories need to be put into practice. Different data types offer various representation options. In this section audio and symbolic features are introduced. Also, the Convolutional Neural Network (CNN) can be used for feature extraction which is explained more precisely in the following text.

2.2.1 Audio features

Audio features have been the heaviest studied kind of features in the field of MER over the last decades [8, 21]. The advantage is that music is usually stored as an audio file from which rich information can be automatically extracted. As there are lots of different features, only some examples are introduced here. Feature spectrograms that are built up by a time and frequency axis make sense in the context of audio files [35]. A few examples are as follows:

- Short-Time Fourier Transform (STFT) [e.g. 35]. The weak point of STFT is that it fails to represent the low and high frequencies as it is calculated in a linear frequency domain [35].
- Chroma STFT [e.g. 15, 35]. Chroma transforms the inputted spectrogram into a spectrogram that represents the 12 semitones. However, low and high-frequency information gets lost [35].
- Mel-frequency Cepstral Coefficient (MFCC) [e.g. 3, 10, 22, 35, 36, 39]. MFCC focuses on the spectral shape. Firstly, the STFT is mapped to a logarithmic scale. Therefore, MFCC accounts for the human perceptual features [30]. Then Discrete Cosine Transform is used on it [21]. It is shown that on the EMOPIA dataset, the MFCC is more strongly correlated with Russell's four emotion quadrants than other commonly used audio features. Therefore, it is evaluated as a valuable model input feature [32].
- Constant Q Transform (CQT) [e.g. 35]. It is determined with the help of a logarithmic frequency domain and honors low and high frequencies [35]. Yang et al. [35] finds out that CQT has an advantage compared to STFT, MFCC, Chroma STFT, and Chroma CQT.
- Cochleogram [e.g. 6, 36, 39]. It mimics the audio signal decomposition of a human cochlea [36].

Beyond the spectral features also other feature types can be extracted from an audio signal. Some instances are the tempo [e.g. 22, 29] and the Zero Crossing Rate (ZCR) [e.g. 3, 22] which counts how often a waveform changes its sign [21].

2.2.2 Symbolic features

Features can also be extracted from symbolic music representation like MIDI files. For three decades the MIDI format has been widely used in the internet and science [19, 30]. It represents the performance of how music is generated and how instruments are active. It is not about the representation of musical sound directly [19].

Features, like the duration of individual notes, pitch in the form of MIDI note values, and major vs. minor mode, can be extracted [36]. In 3.3.1 further features are described which are also used by the underlying thesis.

Beyond that, MIDI files are a time-lined sequence. Having that fact in mind, new feature types can be formulated. For instance, Zeng et al. [37] bring up the idea of splitting the MIDI into single sequence steps. Each sequence step is then transformed into an OctupleMIDI representation which is a tuple containing the following information: time signature, tempo, bar, position, instrument, pitch, duration, and velocity. This representation is fed as one input with its position information into a transformer. The transformer takes all OctupleMIDIs from all time steps as input. Finally, a classifier can work with the hidden outputs of the transformer. The full model is called MusicBERT. Over different music tasks it turns out that the model and also the OctupleMIDI representation style hit other technologies [37].

Chou et al. [4] agree with the findings by showing that the Compound Word (CP) representation beats the revamped MIDI-derived events (REMI) in different musical tasks including the Russell's quadrants prediction. The difference between the representation styles is that CP delivers several symbolic information pieces at one time step whereas REMI has only one information cue per time step. The advantage becomes clear as one-time step content is fed as one unit in the transformer [4].

2.2.3 Further features

Beyond the listed features above other feature types can be used. For example, the song lyrics are quite commonly analyzed in MER [e.g. 1, 3, 5, 8, 11–13, 29, 41]. The network inputs can be made different if lyric features are used. It is possible to process the text using transformers [1] or simply pre-process it before using it as input [5, 13]. Furthermore, high-level features like the genre, gender of the singer, danceability, and dynamic complexity can be considered [21]. Biological features are at the start of their research career [8]. However, those features aren't relevant to the underlying work. Only the genre is indirectly used for prediction by making use of the MusicBERT study [37].

Even further, CNN can be used to find out the important features as it is described in the upcoming section. Because CNN is directly involved in the network's architecture, it is embedded in the text which presents the AI models that are used in the field of MER.

2.3 Common AI models in MER

2.3.1 Models concerning scalar features

When the inputted features are independent of each other and the time-line is ignored then the following models are suited for operations.

Sharma et al. [29] state that for Russell's quadrant prediction the Gaussian Support Vector Machine (SVM) performs the best whereas the decision tree is the worst one in comparison to k-Nearest Neighbors (kNN), Logistic Regression, Naive Bayes, Polynomial SVM, Random Forest (RF), and SVM. However, when the dataset, emotion model, and feature set are changed also other models can be top performers. For instance, Naive Bayes and Logistic Regression can outperform a Multi-Layer Perceptron (MLP) and SVM [22, 24].

2.3.2 CNN based models

In MER research, it is not uncommon to split the network into a feature extracting convolutional part and a downstream classifier part [10]. The CNN automatically identifies

hidden features, like pitch, tempo, loudness, clarity, and rhythm, in two-dimensional input data [26]. The classifier network can be arbitrary. Different researchers use for example the following models: bi-directional Long Short-Term Memory (BiLSTM) or LSTM combined with dense layers (LDNN), dense layers, kNN, RF, and SVM [5, 6, 10, 13, 15, 26, 35, 39]. The kNN, RF, and SVM are outperformed by the LDNN [10] and LSTM [26].

In contrast to the LSTM, the BiLSTM shall be able to represent the emotions in music [6, 39]. It can process the sequence forward and backward and then be able to connect the two outputs [6, 39]. In favor of this, Du, Li, and Gao [6] reach through the CNN-BiLSTM Root Mean Square Error (RMSE) scores very close to 0.

Even further going, the CNN can be followed by a LSTM, a dense layer, and finally, a GAN system which generates fake features out of the dense output on the one hand and on the other hand it takes the output features of the dense layer and separately puts both into a discriminator. This network is named WLDNN_GAN [36]. It outperforms its competitors GAN, kNN, LSTM, and SVM in Mean-Squared Error (MSE) but struggles in beating CLDNN_BiLSTM, and Wavenet in unimodal settings. Also, its R^2 is quite weak compared with other methods whereas SVM was the strongest one [36].

Moreover, the usage of inputs can influence the performance. CNN and DNN architectures can be outperformed by a residual network (Resnet) [33]. The idea of Resnet is to copy the input. One version is the usual network input, the other one can jump over several layers and is then summed up to the layer output. Wang [33] build up the Resnet with convolutional layers. It reaches a better loss score than its blank convolutional equivalent. Also, the idea of Inception is introduced. Here a layer output is copied several times. Each copy gets a one-dimensional convolution with probable pooling and different kernel sizes. The results are concatenated. Then a dense layer follows. However, Inception performs the worst [33]. However, still the kernel design choice can play a crucial role in the CNN performance. For example, Russo et al. [26] observe that a dilated kernel with a bigger receptive field performs the best.

Also worth to mention, less complex networks with fewer nodes and layers can be superior to complex networks because redundant information can be avoided [15].

2.3.3 Multi-modal models

So far, only models dealing with one data source matrix have been discussed. However, MER research is also quite interested in multi-source applications. For instance, several audio features [6, 10], audio and lyric features [5], or even symbolic and audio features [36] might be inputted in the network at once. For example, Huang et al. [11] reach with their bi-modal Deep Boltzmann Machine (DBM) better results when combining lyric features with audio features. This raises the question of how different data sources can be fused. In the following text, different fusion strategies are presented.

The early fusion, which is also known as feature-level fusion, works directly with the data [30]. Features are merged before they enter the network [3]. The fusion is enabled by normalization, transformation to same measure units, machine learning methods, like SVM and CNN, and dimensionality reduction for which features are mapped into a low-dimensional space [30]. An alternative can be the multi-scale fusion algorithm which uses the Inception as it is described in 2.3.2. It targets to consider different feature types by using different kernel sizes. The combination of Resnet with a multi-scale fusion algorithm is called EMOMUSICNET. However, it is hard to train for large data amounts [33]. Going beyond, this approach can be also utilized for mid-level fusion which brings the separate network branches together by concatenating their outputs to continue with a single branch network [5]. It happens at the second or later layer of the network [12].

Late fusion, also known as late-level or decision-level fusion, combines the outputs of several independent algorithms. Here it is possible to give each feature its own best-fitting network. There are several ways to fuse the results. Possibilities include taking the maximum or minimum, combining the outputs in a linear or weighted way, such as the weighted average, and voting [5, 12, 30]. Non-linear transforming fusion reaches best scores [14].

Delbouys et al. [5] show that a bi-modal approach, no matter if late or mid-level fusion, can be superior compared to uni-modal approaches. However, studies may end up in different conclusions. For instance, Krols, Nikolova, and Oldenburg [12] take early fusion into account by fusing audio features with lyrical features. The regressive models Multiple Linear Regression (MLR), MLP, RF (RFR), and SVM (SVR) are compared. For the prediction of valence, all multi-modal models outperform their uni-modal equivalents. However, for the arousal, it was vice versa except the RFR [12]. Catharin et al. [3] and

Sharma et al. [29] agree with these findings of advantage in valence but no one in arousal. In the case of Li et al. [14], valence is predicted the most precisely through a fused SVR. The best performer in arousal prediction is the Artificial Neural Network (ANN) [14].

Also, through the multi-modal setting the performance hierarchy of network types may change. In contrast, to the statement of the superiority of deep learning compared to SVM above [10], for the bi-modal, mid-level fusion case, deep learning is only slightly advantageous compared to late-fusion SVMs. Even further, through lyric input features the SVM hits deep learning approaches like BiLSTM, convolutional layers paired with LSTM, Single Gated Recurrent Unit (GRU), and other similar network types [5].

Even further, studies should consider that there are different mid-level fusion options within one network. Wang et al. [32] observe that the layer of fusion can influence the results. When having a CNN followed by several dense layers, there are different options for where exactly to fuse the hidden features. This can make a drastic difference in model performance [32].

Moreover, the feature extractor Deep BiLSTM (DBLSTM) evaluates audio features from short snippets but also keeps the temporal context in mind [14]. It can be shown that for the case when on a fusion a classifier follows, the lowest RMSE scores are achieved compared to a standalone classifier or fusion, a late-level fusion, and a fusion which happens before and after the classifier. However, all the mentioned methods still perform better than simply not using them [14].

In summary, mid-level fusion is still an eye-catcher. An innovative idea of mid-level fusion is delivered by Zhao et al. [41]. Cross-Modal Attention (CMA) enables hidden feature fusing from several modalities. It is based on a trainable attention technique. The study implements a hierarchical structure. That means first the low-level semantic features are fused. These features can be the song lyrics or a convolutional output, for example. This results in high-level semantic information which can be then fused with the input features of the same sort, e.g. track name and artist. Both the hierarchical fusion strategy and the CMA lead to an increased performance and beat competing models [41].

3 Method

In this chapter, the dataset used and label categorization are introduced in 3.1. Afterward, symbolic data generation is a topic in 3.2. In 3.3 the extraction of symbolic features with the help of the paper of Panda, Malheiro, and Paiva [20] and jSymbolic [16] are examined. Additionally, various packages for audio feature extraction are explored in 3.4. Lastly, the architectures of implemented uni- and multi-modal models are explained in 3.5.

3.1 Dataset principles

3.1.1 Database characteristics

When conducting research, the dataset used can greatly impact the performance of the network [1]. Therefore, it's important to carefully consider the dataset choices. Sometimes, it's necessary to create a new dataset that meets the standards of the research question [10]. Other times, researchers use one or a combination of several datasets to gather more information [15, 35].

Before proceeding with the thesis, it's important to carefully consider the database and label style to be used. This is because AI learns from the input and predicts the labels that should be valid measures of the searched target condition. Taking into account that hat the dataset used can greatly impact the performance of the network, it may be necessary to create a new dataset or use one or a combination of several datasets to gather more information. Because this thesis is part of a university project¹ the needed database with

¹Lab for Personality, Music, and Emotion Psychology of the University of Innsbruck and Institute of Computational Perception of the Johannes Kepler University of Linz,
<https://psychologie-shiny.uibk.ac.at/emma/>

its labels is already fixed. The project encourages the creation of the so-called "Emotion-to-music Mapping Atlas" (EMMA) database.¹ That database arises from the two datasets Music4All-Onion [18] and LFM-2b [27]. The music piece samples are selected for the EMMA only when they have in both datasets an entry marking at least one listening event in LFM-2b. All 370 tracks are freely accessible as mp3 and MIDI files². LFM-2b considers user listening behavior on Last.fm³ in sample selection. It aims to work with micro-genres like the EMMA does [27]. The music genres of EMMA are classic, hip-hop, and pop. The Music4All-Onion dataset represents a multi-modal music dataset considering low-level descriptive audio features (LLD) up to visual video information [18].

The EMMA database uses a unique approach compared to its predecessor datasets. It allows participants to rate each sample according to its GEMS dimension expression, which is a fresh and innovative idea. So far 567 participants have taken part, November 11, 2022.¹

3.1.2 Label categorization

The labels represent the GEMS dimensions are freely accessible online.¹

Due to the nature of the float scalar design of the labels, it offers itself to design a multi-regressive task in which the three GEMS factors, sublimity, vitality, and unease [38], shall be predicted. Furthermore, a multi-classification task can be formulated out of the float-valued labels as follows. A label array for one sample looks like [sublimity value, vitality value, unease value]. The aim is, to mark a significant, regressive label with "1". The minor label expression(s) shall be marked with "0". This is determined by having a look at which factor in the sample reaches the highest value compared with the median of each factor of the whole standardized dataset. The out-performing factor is marked with "1". In addition, when an other factor accomplishes the 65. quantile, it is assumed that the factor is also significant in its expression. In consequence, this factor is also marked with "1".

Looking at all provided samples the threshold quantile of 65 delivered the best label distribution:

- one positive label in 77.53 % of the cases

²<https://fileshare.uibk.ac.at/d/50847c33e13141d1a69a/>

³<https://www.last.fm/>

- two positive labels in 21.37 % of the cases
- [1, 1, 1] in 1.10 % of the cases

Closer distribution information is provided by fig. 3.1. It is assumed to be seldom that all three factors are reflected with the same intensity in one sample that only lasts around half a minute. Thus, the positive fulfillment of all three labels is avoided. Other percentile thresholds are tried out. Increasing the threshold restricts the vast majority of the samples to one positive label. Lowering the threshold leads to an increase in samples for which all three factors are positive. In conclusion, the 65. quantile border is a good representative of the emotional expression of the individual sample.

Beyond that, the comparison base median can be replaced by the mean. The threshold can be defined by the upper standard deviation away from the mean. Doing so the following label distribution appears:

- one positive label in 93.97 % of the cases
- two positive labels in 6.03 % of the cases
- [1, 1, 1] in 0 % of the cases

The thesis works with a small dataset. Hence, music pieces representing two positive labels would appear quite seldom such that the machine learning model could have problems in recognizing them. Apart from this, the weak point of the mean is that outliers have a higher chance than other samples of being selected as positive. The consequences become clear in fig. 3.2 and fig. 3.3. The factors sublimity and vitality are roughly normally distributed without outliers. However, unease represents a right-skewed distribution with lots of outliers in the positive direction. Therefore, the median base was chosen instead of the mean.

So far, this work has dealt with the design of data. More precisely, the questions of symbolic music sample generation and the representation of labels have been answered. In the upcoming section, feature extraction from the finally chosen MIDI files is treated.

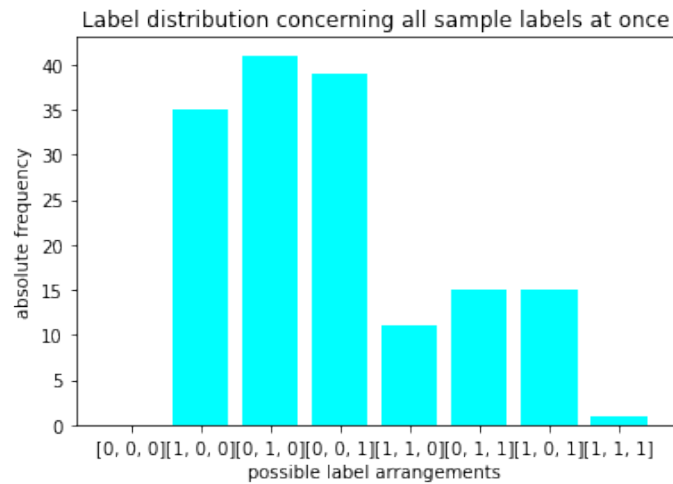


Figure 3.1: Samples as their categorized label combination [sublimity, vitality, unease] which are determined with the help of the median with a threshold percentile of 65.

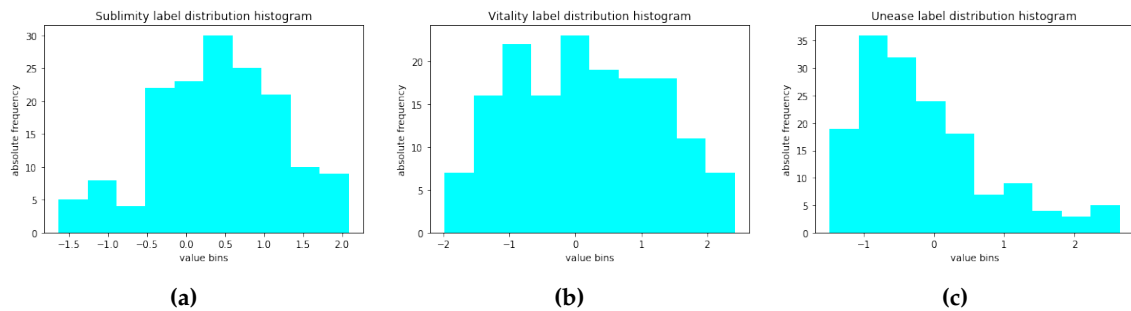


Figure 3.2: Binned value distribution for (a) sublimity, (b) vitality, and (c) unease.

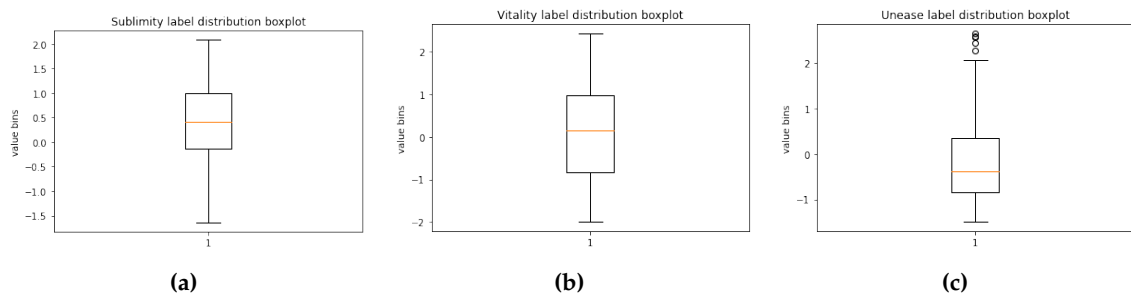


Figure 3.3: Boxplots for (a) sublimity, (b) vitality, and (c) unease.

3.2 MIDI data collection and generation

The well-known, symbolic music representing MIDI format is chosen to be the focus of this work. However, the MIDI files have to be collected or generated first. One aim is, to collect MIDI file data that illustrates all samples of the EMMA database. A variety of approaches are tried out. Those are presented in the following sub-sections.

3.2.1 Web scraping of MIDI files

Download

The initial approach for gathering the required data is to automatically download MIDI files. The goal is to acquire complete music pieces without any cost. To achieve this, the following websites are utilized:

- bitmidi.com
- cprato.com
- freemidi.org
- metal-midi.grahamdowney.com
- mididb.com
- midisfree.com⁴
- miditune.com⁴
- midiworld.com
- nonstop2k.com⁴
- partnersinrhyme.com
- rppmf.com

In the case of classic music pieces the searched titles don't always fit the web titling. Therefore, manual downloading can be done via following web pages:

- kunstderfuge.com⁴
- lvbeethoven.fr

Some web pages have a daily download limit. That can be an obstacle when lots of data needs to be collected. Therefore, a "Virtual Private Network" [31, p. 9] (VPN) can help to overcome the daily download limit by using a new IP address [31]. This work uses the

⁴daily download limit given

Windscribe VPN which can be implemented by installing it on the local computer.⁵ To ensure that no malware is downloaded a viral check is embedded in the web scraping Jupyter Notebook. More precisely, VirusTotal [23] offers a python module virustotal-python⁶. The free version offers a checking limit of 500 local computer files/URLs per day. To make use of it, a user account needs to be created to get access to a personal API key [23]. This key can be used as an ID marker in the web scraping notebook. As a result, the notebook checks for viruses before accessing the web pages and checks again the downloaded MIDI files.

Align MIDI files to their original tracks

With the approach of web scraping MIDI files of whole music pieces are downloaded. However, the EMMA database doesn't build its labels on whole music pieces but only on snippets that last less than one minute. It becomes clear that the downloaded MIDI files can transport more and different GEMS emotions than the snippets. This creates inaccuracy. To address this problem, the downloaded MIDI files are trimmed to match the contents of the mp3 files' snippets, with each sample indicating the start and end time of the snippet.

To begin this task, we need to estimate the location of a snippet within a MIDI file. This involves finding the exact time frame of the mp3 file within the MIDI file. When working with MIDI files, it's important to understand that they are structured symbolically and not measured in seconds. Instead, they are measured in ticks, which allows for precise synchronization and timing within the file. It is significant to consider that the tempi of the mp3 file and the MIDI file can differentiate from each other. The mp3 file information also contains the tempo of the track. The same accounts for the MIDI files. With that knowledge, it is simple to compute the beats per second (bps). It is assumed that 1 bps is equivalent to the length of one quarter per second. By matching the tempi of the two files, the starting second of the snippet within the MIDI file can be computed by $\frac{\frac{1}{\text{bpsMIDIfile}} \cdot \text{startingSecondMP3File}}{\frac{1}{\text{bpsMP3File}}}$. The Python package Mido can transform the starting second information into a tick unit. Through that, precise location information about the start of

⁵download VPN via Ubuntu command shell: <https://www.geeksforgeeks.org/how-to-setup-vpn-on-ubuntu-linux-system-for-ip-spoofing-using-windscribe/>
 official download page: <https://windscribe.net/download>

⁶<https://github.com/dbrennand/virustotal-python>

the snippet within the MIDI file is delivered. The same process is used to determine the end location of the snippet.

For the three mp3 files without tempo and snippet position information, the MIDI files are clipped by their middle parts plus or minus 15 seconds. If a MIDI file's duration is less than or equal to 30 seconds, the entire file is stored as a "snippet file." Besides that, it can happen that a downloaded MIDI file doesn't contain the full version of a music piece. In conclusion, a MIDI file might be shorter than the time locality of the aimed snippet. 33 MIDI files are affected. In this case, the whole MIDI file is stored. When the MIDI file is prepared as input of a model as a note array windows are built in regular distances to represent the piece but in a condensed format.

Next, the cutting process is of interest. MIDI files follow a certain architecture such that it is not as simple as deleting notes. The solution is to play all notes that fall outside of the designated time window with a Mido message velocity of 0. With that strategy, the notes are treated as breaks. As a result, the time snippet is padded by breaks such that a MIDI file keeps its length but loses all information that is not needed. As a consequence, in the upcoming feature selection, all break-related features have to be ignored.

It follows that the snipping process hits some limitations. The MIDI files aren't standardized as they are downloaded from different web pages and are created by different humans. Furthermore, they can differ from the mp3 tracks by focusing on other song versions. For example, it might not be always clear when a music piece starts and ends. Due to the restricted access to enough MIDI files versions are considered that might not be complete or only finalize a music excerpt out of the whole song.

In summary, it appears that there is a limitation to the web scraping method for obtaining music samples. Unfortunately, only 161 MIDI files out of 370 samples are freely accessible, which is less than half of the dataset. However, there is a new strategy being tested, which involves the automatic conversion from audio to MIDI files. More information about this can be found in the upcoming subsection.

3.2.2 audio2MIDI converter

A key for solving the missing MIDI data issue could be to use modules with which the already provided audio files can be automatically converted into MIDI files, in short,

audio2MIDI converter. Several different converters are tried out. According to Wang et al. [34], it is recommended to split music pieces into instrumentals and singing voices and only use the instrumentals as inputs [34]. To do this, the module Spleeter [9] is used.⁷

Basic-Pitch [2]

Basic-Pitch is a module which extracts the CQT from the audio file. The CQT is then used as input of a neural network with convolutional layers. The network can learn to generate two-voice piano MIDI files through training [2]. The pre-trained module can be downloaded charge-free.⁸

The converter of Wang et al. [34]

As an alternative approach, one idea is to use supervised machine learning with an encoder-decoder architecture that incorporates convolution and GRU [34]. This method involves using symbolic files as target labels and a pre-trained pop-song specialized model is freely available.⁹ The resulting MIDI files are simple two-voiced piano files [34]. Due to its results presented in table 4.2, it is chosen to be the basis of the input for the multi-modal model.

Eventually, it is easier to convert the audio files into MIDI files when the audio files follow a simpler structure. Thus, the single samples are split with the help of Spleeter into their single voice tracks. Only the singing voice gets ignored. For each voice track, Wang et al. [34] converter generates one MIDI file. Afterward, through the Mido module¹⁰ the different voice tracks can be simply added to each other such that the final MIDI file consists of all music tracks in piano style. Listening to those songs they clearly distinguish from the originally produced MIDI file. However, this approach doesn't lead to better results.

⁷<https://github.com/deezer/spleeter>

⁸<https://github.com/spotify/basic-pitch>

⁹<https://zzun.app/repo/ZZWaang-audio2midi>

¹⁰<https://github.com/mido/mido>

Pearce's transcriptor

Pearce's STFT approach¹¹. With MATLAB STFT is extracted. It shall be transformed into MIDI. However, the results are sobering. For the human ear, the original music piece has nothing to do with the generated MIDI piano sound. As it doesn't bring added value, this converter isn't considered in the results.

To sum it up, there exist a bunch of audio2MIDI transcriptors. The tried-out approaches produce MIDI files that at first glance don't have a lot in common with the original sound. However, how well they perform is presented in 4.

3.3 Feature extraction for symbolic data

The basis of this thesis involves working with symbolic music representation. As a result, it is important to extract features that align with this structure. These features are known as high-level features as they encompass musicological and music-theoretical knowledge that may not be readily apparent in audio files. For instance, identifying instrument types in audio files can be challenging, whereas MIDI files encode instrument names directly [17]. In the following section, it is explored how various types of features can be extracted from a symbolic representation.

3.3.1 Manual feature extraction inspired by Panda, Malheiro, and Paiva [20]

The aim is to consider all voice tracks within each music piece. It is implemented to treat each voice separately or to consider them all equally. The latter is finally used. The following concepts originate from the paper of Panda, Malheiro, and Paiva [20]. The authors deal with the extraction of features that can be important for the music emotion understanding. The features, that fit the underlying symbolic data and topic of the thesis, are collected. The paper states that musical features are categorized into eight groups: melody, dynamics, rhythm, musical texture, and expressive techniques like articulation, glissando, and vibrato. Also, harmony, tone color, and musical form are such dimensions, but they are not of interest in this sub-chapter [20].

¹¹<https://github.com/TeaPearce/Audio2Midi>

Generally speaking, Melodic Features concern pitch information. In the symbolic music representation context a pitch represents a note's MIDI value. The following features can be determined:

- basic statistics regarding the pitch: maximum and minimum
- "Note Smoothness statistics" [20, p. 8]: take the pitch mean, standard deviation, skewness, kurtosis, maximum, and minimum but instead of using individual MIDI notes the absolute difference between two consecutive notes is taken.
- The note value difference between successive notes is categorized into lower, equal, and higher. Then, $\frac{\text{category}}{\text{TotalNoteNumber}-1}$ ratios are computed.
- note values are separated into classes Soprano, Mezzo-soprano, Contralto, Tenor, Baritone, and Bass. Afterwards, the ratios $\frac{\text{class}}{\text{TotalNoteNumber}}$ are taken.

Moreover, the dynamic features are interesting in the context of symbolic music. They concern themselves with note intensity and its variation. First, the question arises of how the salience of a note can be defined in the symbolic context. In general, salience describes the human perception of a played pitch [21]. In the event of symbolic music representation, the velocity gives information about the intensity of a sound. It represents the volume for a note-on MIDI message which marks the start of a note. In case of a note-off message, which sets the end to a currently played note, velocity can be described as note decay [19]. The higher the velocity is, the higher the salience scores. Through Mido, the note salience can be simply computed by taking the median of the velocities of the note-on and note-off messages of the individual note.

From these considerations, the following features can be formulated.

- basic statistics referring to the note salience: mean, standard distribution, skewness, kurtosis, maximum, and minimum
- the computation of intensity by comparing the individual note salience considering the mean salience with half standard deviation. From this follows the note intensity classes low, medium, and high. In consequence, the ratios $\frac{\text{IntensityClass}}{\text{TotalNoteNumber}}$ are computed.
- note intensity difference between successive notes are categorized into lower, equal, and higher. Then, ratios $\frac{\text{class}}{\text{TotalNoteNumber}-1}$ are computed.

- Finally, the "Crescendo and Decrescendo" [20, p. 11] (CD) are of interest. Here the first and second parts of the individual note are used to determine the intensity difference. A variation of 20 % of the median is needed to be considered as CD. Even further, the sequence length of CD notes of the same type and the six statistics are computed.

The rhythmic features consider sound patterns and silences within a music piece [21]. Using this knowledge the tailing features can be created.

- basic statistics of the individual note duration values: mean, standard distribution, skewness, kurtosis, maximum, and minimum
- Consecutive notes are analyzed by looking at the change in raw note duration between them. A change is crucial when the length difference is at least 10 % and can be therefore titled with shorter or longer. Otherwise, the changing class is called "equal". Then, the fractions $\frac{\text{ChangingClass}}{\text{TotalNoteNumber}-1}$ are computed.
- Short, medium, and long note duration is formulated through comparison with the note length mean value with half standard deviation. From this follow the ratios $\frac{\text{DurationClass}}{\text{TotalNoteNumber}}$.

The last feature group of interest is the expressivity which discusses ornaments [21].

- the ratios of staccato, legato, and other transitions
- The ratios are also computed for the blank note duration for each of the three transition types, more precisely $\frac{\text{SumNoteDurationsOfOneTransitionType}}{\text{SumAllNoteDurations}}$.

This section presented different features to extract in bullet points. However, this list is far from being complete. The next section makes it possible to extract even more symbolic features from MIDI files.

3.3.2 jSymbolic

jSymbolic is an open-source, Java-based software that has been further developed over the years [16, 17]. The underlying thesis uses jSymbolic 2.2 which extracts 246 different features from the MIDI files [16]. Only the scalar features are of interest. Those concentrate

on the computational measures of chords, dynamics, instruments, meter, notes, note densities, pitch, rhythm, tempo, triads, and voice tracks [16].

3.4 Feature extraction for audio data

Low-level audio features, may not have an immediate musical quality, but are highly useful for machine processing [17]. These features can be identified by analyzing the audio files. In the next section, various types of audio features will be introduced.

3.4.1 Statistical features

The first audio feature group simply describes statistical measures. Rhythmic, tonal, and low-level feature statistics, more precisely the mean, standard deviation, variance, median, minimum, and maximum, are chosen. With the package Essentia¹² standard mode, it is possible to compute them.

3.4.2 Functional features

Even further, the group of functional features is of interest. The packages emobase and Computational Paralinguistics challengeE (ComParE) of OpenSMILE¹³ can compute them as scalar and time-windowed statistical features. For the latter case, the module openXBOW uses a Bag of Audio words approach (BoAw) to limit the feature number to 250 "words" [28]. Both, emobase and ComPare, carry part-wise the same feature names with quite different values. Therefore, both versions are selected to end up in the final model input.

Besides that, scalar functional features are also outputs. In the case of emobase, that are the fundamental frequency (F0) and its envelope, pcm intensity, and pcm loudness. ComParE determines audspec, audspecRasta, pcm RMSenergy, pcm zcr, pcm fftMag, F0final, jitterLocal, and logHNR. The features that help to capture temporal patterns in the data, are MfCC and lspFreq from emobase, as well as audSpec Rfilt and MfCC from

¹²<https://essentia.upf.edu/download.html>

¹³<https://github.com/audeering/opensmile>

ComParE. It is worth mentioning that the "voice"-related features are ignored as the thesis works with samples that are separated in instrumentals and the unused singing voice.

3.4.3 LLD features

Even further, emobase computes low-level descriptors (LLD) which are in the shape of time frames x features. The LLD is converted through openXBOW into 250 "words".

3.5 Models

In section 2.3, the relevant machine learning ideas are covered that influence the designs of the models in the underlying thesis. The upcoming sub-section introduces these models in more detail.

3.5.1 Uni-modal models

First of all, it is reasonable to mitigate the complexity of the training process. Therefore, the most important audio and symbolic features can be selected through the modules RFE and SelectFromModel of the package `sklearn.feature_selection`¹⁴. For each package the number of top features is set to 20. Because both find part-wise same features, the total number of best audio and symbolic features results in 30. Those are listed in the table 4.1. The underlying inputs are the audio features and the symbolic features extracted from the web scraped snippets. Regressive labels are used. The top 30 features which are introduced here are found by usage of the regressive labeled data.

With the help of scikit-learn¹⁵ different models can be considered, namely kNN, RF, and SVM. Each model gets its own pipeline for the selection of the best hyper-parameter setting. For the adjustment of this idea, the training is executed by cross-validation in the form of a stratified shuffle split. Thus, the dataset is split into 20 % for the test set, and 80 % for the training set from which 20 % are needed for the validation set.

¹⁴https://scikit-learn.org/stable/modules/feature_selection.html

¹⁵<https://scikit-learn.org/stable/index.html>

3.5.2 Multi-modal model architectures

The multi-modal model shall make it possible to use the input of different data sources. From the audio signal, the mel-filtered STFT is chosen to be the input. For MIDI files generated with the transcriptor of Wang et al. [34] the collected notes of each voice in the form of a two-dimensional matrix can be used. As an alternative, the pre-trained model of Zeng et al. [37] with its OctupleMIDI representation can be deployed as it is more detailed explained in section 2.2.2. The small version of the pre-trained genre predicting model is used to handle the complexity of the training task. Besides that, the same top 30 features of the section 3.5.1 serve as input.

The model architecture is visualized in fig. 3.4. The graphic process direction goes from the left to the right. The legend's variables also appear in the final source code. The `conv_mode` fixes whether the use of convolution in the audio track is wanted. The variable `musicbert_mode` specifies whether the approach of Zeng et al. [37] is re-used or the note array is taken for the symbolic input.

Paolizzo et al. [22] draw attention to the fact that the underlying problem is both a multi-label problem and a multi-class problem. Multi-label becomes active through the fact that annotators can use multiple emotion labels for one track. Several can be present, but there is no exclusivity. Multi-class happens to be the case as each emotion can be part or not of the sample [22]. Therefore, additional dense layers are established to concentrate only on one label to predict. If the task is formulated as binary classification for each label, the final prediction supplies two output nodes and is trained with Binary Cross Entropy (BCE) with logits loss for each branch. This is given when the variable `label_type_categorical` is set to `True`.

For a regressive prediction setting, it is tried to predict a concrete value that gets trained on the metric MSE. For this purpose, the outputs are inputted into a tanh activation function and then multiplied by 3 to mimic the distribution of the ground truth as it is visualized in fig. 3.2.

Stratified, shuffled cross-validation enables the hyper-parameter search. The test set takes 20 % of the data amount, and from the remaining training data, 20 % are taken for the validation set.

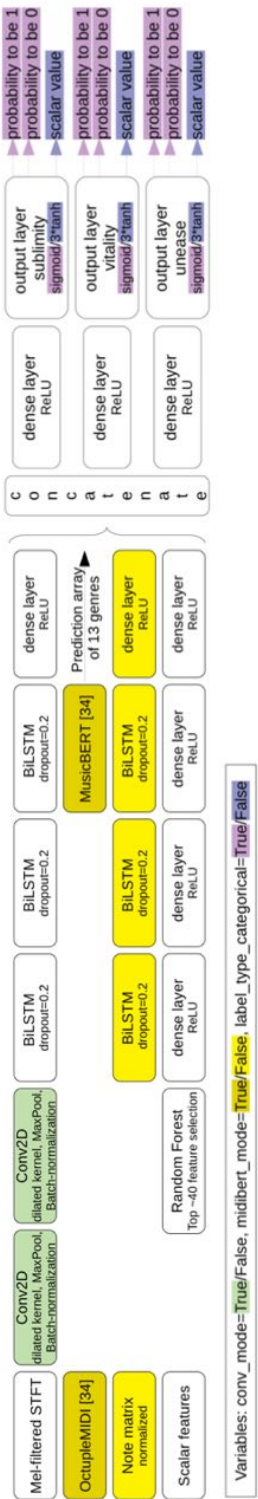


Figure 3.4: Visualization of the used multi-modal model which starts with the inputs (left) and ends up with the outputs (right). It depends on the variables in the legend box how the architecture looks in detail.

4 Results

4.1 The top 30 scalar features

In sub-section 3.5.1 the top 30 scalar features are introduced. Those are all listed in table 4.1. It is noticeable that only one symbolic MIDI feature is among them.

4.2 Performance of the uni-modal model

The table 4.2 indicates that no solution strategy dominates the other ones. Each model makes only one vs. other label predictions. That means that for each label a separate estimator is needed. With a look at the table, it turns out that the results are far from being perfect considering this condition. Moreover, no data source outperforms obviously the others. On the categorical label condition, the web scraped full songs, web scraped snippets, and the audio2midi Basic-Pitch converter [2] reach a rating of at least 60 % accuracy five, five, and six times. However, with consideration of the regressive labels, the viewpoint changes a bit. In this case, only the audio2midi converter of Wang et al. [34] can hold foot with the web scraped snippets.

To complete the results the hyper-parameters for the estimators of the table 4.2 are listed in the related table 4.3.

audio	MIDI
audio_F0_sma_maxPos audio_F0final_sma_de_flatness_f2 audio_audspec_lengthL1norm_sma_de_iqr1-2_f2 audio_audspec_lengthL1norm_sma_de_iqr1-3_f2 audio_audspec_lengthL1norm_sma_de_iqr2-3_f2 audio_audspec_lengthL1norm_sma_de_quartile1_f2 audio_audspec_lengthL1norm_sma_de_stddevRisingSlope_f2 audio_audspec_lengthL1norm_sma_de_upleveltime50_f2 audio_audspec_lengthL1norm_sma_iqr2-3_f2 audio_audspec_lengthL1norm_sma_lpc3_f2 audio_jitterDDP_sma_linregc2_f2 audio_logHNR_sma_de_lpgain_f2 audio_logHNR_sma_lpc4_f2 audio_lowlevel.dissonance.mean audio_lowlevel.spectral_entropy.mean audio_pcm_fftMag_spectralEntropy_sma_de_iqr1-2_f2 audio_pcm_fftMag_spectralEntropy_sma_de_iqr2-3_f2 audio_pcm_fftMag_spectralEntropy_sma_de_quartile3_f2 audio_pcm_fftMag_spectralRollOff90.0_sma_leftctime_f2 audio_pcm_fftMag_spectralSkewness_sma_lpc2_f2 audio_pcm_fftMag_spectralSkewness_sma_peakDistStddev_f2 audio_pcm_fftMag_spectralSlope_sma_de_maxSegLen_f2 audio_pcm_fftMag_spectralSlope_sma_de_minRangeRel_f2 audio_pcm_fftMag_spectralSlope_sma_de_quartile2_f2 audio_pcm_fftMag_spectralVariance_sma_de_iqr2-3_f2 audio_shimmerLocal_sma_de_upleveltime90_f2 audio_tonal.hpcp_entropy.mean audio_tonal.hpcp_entropy.stdev audio_tonal.tuning_diatonic_strength	midi_Average Note Duration

Table 4.1: Top 30 k using the tools RFE and SelectFromModel of the package sklearn.feature_selection where each tool searches independently the top 20 features based on the regressive labeled MIDI files containing audio and symbolic scalar features.

		Categorical labels (accuracy in %)				Regressive labels (R ² by sklearn)			
data source	model	sublimity	vitality	unease	sublimity	vitality	unease		
Webscraped full songs	kNN	65.63	62.50	56.26	-0.0925	-0.0108	-0.0634		
Webscraped snippets		62.50	43.75	68.75	-0.0437	+0.0063	-0.1930		
audio2midi converter [2]		63.89	61.11	54.17	+0.1015	-0.1187	-0.0216		
audio2midi converter [34]		63.89	55.56	55.56	+0.1336	+0.0138	+0.0445		
audio2midi converter [34]		56.94	58.33	63.89	-0.0203	-0.0605	+0.0251		
fused snippets									
Webscraped full songs	SVM	56.25	43.75	75.00	-0.0338	+0.1289	-0.0352		
Webscraped snippets		62.5	68.75	68.75	+0.1295	+0.0007	+0.0262		
audio2midi converter [2]		61.11	56.94	66.67	+0.0702	+0.0278	-0.1329		
audio2midi converter [34]		63.89	55.56	55.56	-0.0113	+0.0079	-0.0468		
audio2midi converter [34]		58.33	61.11	58.33	-0.0049	-0.0294	-0.1297		
fused snippets									
Webscraped full songs	RF	47.92	62.50	62.50	-0.4178	+0.0569	+0.1753		
Webscraped snippets		58.33	50.00	47.92	+0.1240	+0.4103	-0.1758		
audio2midi converter [2]		50.00	72.22	63.89	-0.2099	-0.3789	-0.4560		
audio2midi converter [34]		63.89	52.78	50.00	-0.0409	-0.1016	-0.2350		
audio2midi converter [34]		59.72	58.33	50.00	-0.2069	-0.4381	-0.2542		
fused snippets									

Table 4.2: The results of the best performing uni-modal estimators on the top 30 features considering the different data sources.

data source	Categorical labels			Regressive labels		
	model	sublimity	vitality	unease	sublimity	vitality
Web scraped full songs	kNN	11,distance	1,uniform	31,distance	11,uniform	11,uniform
Web scraped snippets		81,uniform	41,uniform	21,uniform	11,uniform	61,uniform
audio2midi converter [2]		51,uniform	51,uniform	21,uniform	91,distance	11,uniform
audio2midi converter [34]		31,uniform	11,uniform	11,distance	21,uniform	41,uniform
audio2midi converter [34]	SVM	31,uniform	31,uniform	51,uniform	11,uniform	11,uniform
fused snippets		2,scale,rbf	0.9,scale, σ	22,scale,rbf	0.9,auto,rbf	0.5,scale, linear
Web scraped full songs		72,scale,rbf	22,scale,rbf	22,scale,rbf	12,scale,rbf	22,auto,rbf
Web scraped snippets		0.5,scale, σ	52,scale,rbf	0.5,scale, σ	2,auto,rbf	2,scale,rbf
audio2midi converter [2]	RF	0.9,scale, σ	72,scale, σ	72,scale,rbf	0.1,scale, σ	2,scale,rbf
audio2midi converter [34]		92,scale,rbf	22,scale,rbf	2,scale,rbf	0.1,scale,rbf	0.1,scale, σ
audio2midi converter [34]		50,entropy	50,gini	50,entropy	70, error	70, error
fused snippets		50,entropy	50,entropy	50,entropy	230, error	250, error
Web scraped full songs	RF	170,entropy	50,gini	70,entropy	190, error	190, error
Web scraped snippets		50,entropy	50,entropy	70,entropy	50, error	150, error
audio2midi converter [2]		50,entropy	50,entropy	50,gini	50, error	190, error
audio2midi converter [34]		50,entropy	50,entropy	50,gini	50, error	190, error
audio2midi converter [34]	RF	50,entropy	50,entropy	50,gini	50, error	190, error
fused snippets		50,entropy	50,entropy	50,gini	50, error	190, error
Web scraped full songs		50,entropy	50,entropy	50,gini	50, error	190, error
Web scraped snippets		50,entropy	50,entropy	50,gini	50, error	190, error

Table 4.3: Summary of the used hyper-parameter settings whose estimators' results are collected in table 4.2. kNN: number of neighbors and weights; SVM: C, gamma, and kernel; RF: number estimators and criterion.

4.3 Performance of the multi-modal model

midibert_mode	conv_mode	Categorical labels (accuracy in %)	Regressive labels (R^2 by PyTorch)
True	True	62.04	-0.0109
True	False	62.04	-0.0004
False	True	62.04	+0.0019
False	False	62.04	-0.0119

Table 4.4: Best performing multi-modal models on the top 30 scalar features concerning the test set based on the data sources of the automatically generated MIDI files [34] and the music tracks.

The multi-modal model considers the audio tracks and the automatically generated MIDI files by the transcriptor of Wang et al. [34]. The table 4.4 lists the found results based on the test set. It varies the variables `conv_mode` that decides over the usage of convolutional layers, and `midibert_mode` that chooses between the pre-trained MusicBERT model [37] and the BiLSTM path with the MIDI note array as input. The concrete model is visualized by fig. 3.4. Eye-catching is the fact that the variables for architecture choice don't seem to influence the result for categorical labeled data, 62.04 %.

All models no matter what label type have following hyper-parameters in common: `lr=1e-05`, `batch_size=64`, `number_epochs=800`, `betas=(0.98, 0.999)` that relate to the Adam optimizer. The hyper-parameter setting appears to be the same across the variants `midibert_mode=True` in categorical case and `midibert_mode=True` and `conv_mode=True` for regressive labels: `kernel_size_stft=(3, 3)`, `channel_size_stft=6`, `hidden_dim_stft=56`, `out_dim_stft=0`, `bi_direct_stft=True`, `dense_out_stft=100`, `hidden_dim_scalar=750`, `out_dim_scalar=10`, `dense_out_scalar=200`, `final_hidden_dense_out=100` that gives the output size of the dense layer which takes the concatenated output of the different branches. The term "stft" relates to the audio branch whereas "scalar" refers to the branch with the scalar input features. "hidden" indicates the output size of hidden layers. "dense_out" informs about the output size of the individual data source branches. "bi_direct" relates always to the bi-directional boolean question of BiLSTM.

The hyper-parameter setting for the variable version `midibert_mode=False` in the categorical label setting and for `midibert_mode=False` and `conv_mode=False` in the regressive case is as follows: `kernel_size_stft=(3, 3)`, `channel_size_stft=6`, `hidden_dim_stft=56`, `out_dim_stft=0`, `bi_direct_stft=True`, `dense_out_stft=100`, `dense_out_note=56`,

hidden_dim_note=56, out_dim_note=0, bi_direct_note=True, hidden_dim_scalar=750, out_dim_scalar=10, dense_out_scalar=200, final_hidden_dense_out=100. The term "note" is aligned to the MIDI data branch.

The regressive label case for midibert_mode=True and conv_mode=False shows the following setting: hidden_dim_stft=56, out_dim_stft=0, bi_direct_stft=True, dense_out_stft=100, hidden_dim_scalar=750, out_dim_scalar=10, dense_out_scalar=200, final_hidden_dense_out=50.

Finally, the variant midibert_mode=False and conv_mode=True for regressive labels is given: kernel_size_stft=(3, 3), channel_size_stft=6, hidden_dim_stft=56, out_dim_stft=0, bi_direct_stft=True, dense_out_stft=100, dense_out_note=56, hidden_dim_note=56, out_dim_note=0, bi_direct_note=True, hidden_dim_scalar=750, out_dim_scalar=10, dense_out_scalar=200, final_hidden_dense_out=50.

5 Discussion

5.1 Overview with limitations

The underlying thesis generates a MIDI database and implements strong ideas from the current research field of MER. On the one hand, uni-modal models kNN, RF, and SVM based on audio and symbolic scalar features are discussed. They seem to be superior compared to the multi-modal approach. However, they have the advantage that only one out of three labels is predicted per model. The multi-modal machine learning approach considers all three labels at once. For the categorical labeled data it achieves moderate results. However, the multi-modal model struggles in the R^2 measure for regressive labeled data. To push the model to better performance alternative branches are added. For the mel-filtered STFT audio path convolutional layers are added. The note array branch can be completely replaced by MusicBERT [37] predictions. However, no striking performance increase happens.

Reasons for these observations could be that the best hyper-parameter setting isn't found. Especially, for the multi-modal model with its high amount of hyper-parameters regarding the learning process and network architecture a hyper-parameter grid achieves very fast a massive amount of combinations in a long-lasting training process. Furthermore, the implemented models could be too small to achieve huge success. It is also possible that the quality of MIDI data is too low. Very conspicuous is the fact that the used dataset is quite small. 404 audio tracks are achievable but only 369 annotations are provided. Even more violent, the web scraped MIDI files comprise 161 MIDI files. On the other hand, automatically generated MIDI files suffer in quality. Therefore, it seems to be a trade-off between data quality and quantity.

5.2 Limitations in the MER research field

Having a look into the MER research field it is conspicuous that in the most cases a dataset with western music styles is used [6, 8, 26, 33]. Catharin et al. [3] claims that in MER mostly English, Chinese, and Indian music directions are of interest. Even further, the datasets can miss crucial information due to the small size [8]. For instance, Hizlisoy, Yildirim, and Tufekci [10] only predict three out of four valence-arousal quadrants because data for the last quadrant is not available.

Research in handcrafting features in MER still struggles with too small datasets and too concrete genres [21]. Convolutional and recurrent feature representation techniques give no information on which of the original features are essential for which emotion [21]. In general, the field of feature extraction is far from being exhausted [8].

Also, poor label quality isn't uncommon [8]. It can originate from the cognitive load participants are exposed to [7]. Moreover, Yang et al. [35] criticizes that perceived emotions could be dependent on the individual person and be therefore quite subjective. It is difficult to quantify subjective emotions. Emotion perception can vary between different persons and contexts [8]. Also, the labeling studies recruit most of the time western, educated listeners from industrialized, democratic countries that are not exposed to poverty [7].

Even further, it is hard to compare the different MER studies. From above it becomes clear that they use different datasets, emotion models for labeling, evaluation metrics, and different kinds of input features in combination with different network types. Russo et al. [26] also criticizes the various researchers, the varying lengths of sample clips, and the generation of research groups' own databases. In addition, there are no standardized annotation strategies given [7]. Also, copyright laws complicate the usage of free datasets [7].

Over the years, there seems to be only creeping progress in the field of MER. Even classifying tasks tend to stagnant results [8, 21]. Therefore, Han et al. [8] suggest that MER is still in an early research stage.

5.3 Conclusions and future work

The MER research field is far from being exhausted. From above it becomes clear that studies for the dataset creation, feature research, design of deep learning methods, and label annotations are still all needed. A dataset that consists of music styles from all around the world in a representative manner could be the basis for feature development studies. Building on that the machine learning models could be further investigated. However, building up such a massive database that is freely accessible seems to be impossible due to the copyright laws of different nations.

Furthermore, from the struggles described above, it becomes clear that working with a larger, qualitative MIDI database could be advantageous. For this purpose, it could be possible to first collect MIDI files and then annotate them with GEMS labels in experiments. Of course, researchers should also think about the genre and label distributions such that the dataset can be representative. Gómez-Cañón et al. [7] suggest asking cultural experts to choose adequate music pieces.

To solve the problem of subjective annotators, one possible solution option could be to select participants with similar characteristics, e.g. personality and music experience, and attach them to the same group which supplies the ground truth emotion labels. With the help of different groups several ground truths can be investigated [7]. This has the advantage, that the statements in MER are less generalized.

Bibliography

- [1] Yudhik Agrawal, Ramaguru Guru Ravi Shanker, and Vinoo Alluri. “Transformer-Based Approach Towards Music Emotion Recognition from Lyrics”. In: *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, 2021, pp. 167–175. DOI: 10.1007/978-3-030-72240-1_12.
- [2] Rachel M. Bittner et al. *A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation*. 2022. DOI: 10.48550/arXiv.2203.09893. arXiv: 2203.09893 [cs.LG].
- [3] Leonardo Gabiato Catharin et al. “Multimodal Classification of Emotions in Latin Music”. In: *IEEE International Symposium on Multimedia, ISM 2020, Naples, Italy, December 2-4, 2020*. IEEE, 2020, pp. 173–180. DOI: 10.1109/ISM.2020.00038.
- [4] Yi-Hui Chou et al. *MidiBERT-Piano: Large-scale Pre-training for Symbolic Music Understanding*. 2021. DOI: 10.48550/arXiv.2107.05223. arXiv: 2107.05223 [cs.LG].
- [5] Rémi Delbouys et al. “Music Mood Detection Based on Audio and Lyrics with Deep Neural Net”. In: *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, Sept. 2018, pp. 370–375. DOI: 10.5281/zenodo.1492427.
- [6] Pengfei Du, Xiaoyong Li, and Yali Gao. “Dynamic Music emotion recognition based on CNN-BiLSTM”. In: *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*. 2020, pp. 1372–1376. DOI: 10.1109/ITOEC49072.2020.9141729.
- [7] Juan Sebastián Gómez-Cañón et al. “Music Emotion Recognition: Toward new, robust standards in personalized and context-sensitive applications”. In: *IEEE Signal Processing Magazine* 38.6 (2021), pp. 106–114. DOI: 10.1109/MSP.2021.3106232.
- [8] Donghong Han et al. “A Survey of Music Emotion Recognition”. In: *Front. Comput. Sci.* 16.6 (Dec. 2022). DOI: 10.1007/s11704-021-0569-4.

- [9] Romain Hennequin et al. "Spleeter: a fast and efficient music source separation tool with pre-trained models". In: *Journal of Open Source Software* 5.50 (2020), p. 2154. DOI: 10.21105/joss.02154.
- [10] Serhat Hizlisoy, Serdar Yildirim, and Zekeriya Tufekci. "Music emotion recognition using convolutional long short term memory deep neural networks". In: *Engineering Science and Technology, an International Journal* 24.3 (2021), pp. 760–767. DOI: <https://doi.org/10.1016/j.jestch.2020.10.009>.
- [11] Moyuan Huang et al. "Bi-Modal Deep Boltzmann Machine Based Musical Emotion Classification". In: *Artificial Neural Networks and Machine Learning – ICANN 2016*. Ed. by Alessandro E.P. Villa, Paolo Masulli, and Antonio Javier Pons Rivero. Cham: Springer International Publishing, 2016, pp. 199–207.
- [12] Tibor Krols, Yana Nikolova, and Ninell Oldenburg. *Multi-Modality in Music: Predicting Emotion in Music from High-Level Audio Features and Lyrics*. Feb. 2023. DOI: 10.48550/arXiv.2302.13321.
- [13] Sitdhibong Laokok and Subhorn Khonthapagdee. "Emotion Classification in Thai music using Convolutional Neural Networks". In: *2022 6th International Conference on Information Technology (InCIT)*. 2022, pp. 148–151. DOI: 10.1109/InCIT56086.2022.10067398.
- [14] X. Li et al. "DBLSTM-based multi-scale fusion for dynamic emotion prediction in music". In: *2016 IEEE International Conference on Multimedia and Expo (ICME)*. Los Alamitos, CA, USA: IEEE Computer Society, July 2016, pp. 1–6. DOI: 10.1109/ICME.2016.7552956.
- [15] Xin Liu et al. "CNN based music emotion classification". In: *ArXiv abs/1704.05665* (2017).
- [16] Cory McKay, Julie Cumming, and Ichiro Fujinaga. "JSYMBOLIC 2.2: Extracting Features from Symbolic Music for use in Musicological and MIR Research". In: *Proceedings of the 19th International Society for Music Information Retrieval Conference* (Paris, France). Paris, France: ISMIR, Sept. 2018, pp. 348–354. DOI: 10.5281/zenodo.1492421.
- [17] Cory McKay and Ichiro Fujinaga. "JSymbolic: A feature extractor for MIDI files". In: *International Computer Music Conference, ICMC 2006* (Jan. 2006).

- [18] Marta Moscati et al. “Music4All-Onion – A Large-Scale Multi-Faceted Content-Centric Music Recommendation Dataset”. In: *Proceedings of the 31st ACM International Conference on Information Knowledge Management*. CIKM '22. Atlanta, GA, USA: Association for Computing Machinery, 2022, pp. 4339–4343. DOI: 10.1145/3511808.3557656.
- [19] Meinard Müller. *Fundamentals of Music Processing: Using Python and Jupyter Notebooks*. 2nd ed. Cham: Springer, 2021. DOI: 10.1007/978-3-030-69808-9.
- [20] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. “Novel Audio Features for Music Emotion Recognition”. In: *IEEE Transactions on Affective Computing* 11 (2020), pp. 614–626. DOI: 10.1109/TAFFC.2018.2820691.
- [21] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. “Audio Features for Music Emotion Recognition: A Survey”. In: *IEEE Trans. Affect. Comput.* 14.1 (Jan. 2023), pp. 68–88. DOI: 10.1109/TAFFC.2020.3032373.
- [22] Fabio Paolizzo et al. “A New Multilabel System for Automatic Music Emotion Recognition”. In: *2021 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)* (2019), pp. 625–629. DOI: 10.1109/MetroInd4.0IoT51437.2021.9488537.
- [23] Peng Peng et al. “Opening the Blackbox of VirusTotal: Analyzing Online Phishing Scan Engines”. In: *Proceedings of the Internet Measurement Conference*. IMC '19. Amsterdam, Netherlands: Association for Computing Machinery, 2019, pp. 478–485. DOI: 10.1145/3355369.3355585.
- [24] V. R Revathy and Anitha S. Pillai. “Multi-class classification of song emotions using Machine learning”. In: *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*. 2022, pp. 2317–2322. DOI: 10.1109/ICACITE53722.2022.9823535.
- [25] James Russell. “A Circumplex Model of Affect”. In: *Journal of Personality and Social Psychology* 39 (Dec. 1980), pp. 1161–1178. DOI: 10.1037/h0077714.
- [26] Mladen Russo et al. “Cochleogram-based approach for detecting perceived emotions in music”. In: *Information Processing Management* 57.5 (2020), p. 102270. DOI: <https://doi.org/10.1016/j.ipm.2020.102270>.

- [27] Markus Schedl et al. "LFM-2b: A Dataset of Enriched Music Listening Events for Recommender Systems Research and Fairness Analysis". In: *Proceedings of the 2022 Conference on Human Information Interaction and Retrieval*. CHIIR '22. Regensburg, Germany: Association for Computing Machinery, 2022, pp. 337–341. DOI: 10.1145/3498366.3505791.
- [28] Maximilian Schmitt and Björn Schuller. "openXBOW – Introducing the Passau Open-Source Crossmodal Bag-of-Words Toolkit". In: *Journal of Machine Learning Research* 18.96 (2017), pp. 1–5.
- [29] Hardik Sharma et al. "A New Model for Emotion Prediction in Music". In: *2020 6th International Conference on Signal Processing and Communication (ICSC)* (2020), pp. 156–161. DOI: 10.1109/ICSC48311.2020.9182745.
- [30] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. "Multimodal Music Information Processing and Retrieval: Survey and Future Challenges". In: *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*. 2019, pp. 10–18. DOI: 10.1109/MMRP.2019.00012.
- [31] Haydar Teymurlouei and Vareva Harris. "Effective Methods to Monitor IT Infrastructure Security for Small Business". In: *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2019, pp. 7–13. DOI: 10.1109/CSCI49370.2019.00009.
- [32] Hongfei Wang et al. "Emotional Quality Evaluation for Generated Music Based on Emotion Recognition Model". In: *2022 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. 2022, pp. 1–6. DOI: 10.1109/ICMEW56448.2022.9859459.
- [33] W. Wang. "CNN based music emotion recognition". In: *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*. Los Alamitos, CA, USA: IEEE Computer Society, Nov. 2021, pp. 190–195. DOI: 10.1109/ICAICE54393.2021.00044.
- [34] Ziyu Wang et al. *Audio-to-symbolic Arrangement via Cross-modal Music Representation Learning*. 2022. arXiv: 2112.15110 [cs.SD].
- [35] Pei-Tse Yang et al. "Predicting Music Emotion by Using Convolutional Neural Network". In: *HCI in Business, Government and Organizations*. Ed. by Fiona Fui-Hoon Nah and Keng Siau. Cham: Springer International Publishing, 2020, pp. 266–275.

- [36] Lanqing Yin, Jiandong Tang, and Jinming Yu. "Multimodal Music Emotion Recognition based on WLDNN_{GAN}". In: *2022 International Symposium on Advances in Informatics, Electronics and Education (ISAIEE)*. 2022, pp. 528–532. DOI: 10.1109/ISAIEE57420.2022.00114.
- [37] Mingliang Zeng et al. "MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training". In: *ACL-IJCNLP 2021*. June 2021.
- [38] Marcel Zentner, Didier Grandjean, and Klaus Scherer. "Emotions Evoked by the Sound of Music: Characterization, Classification, and Measurement". In: *Emotion (Washington, D.C.)* 8 (Sept. 2008), pp. 494–521. DOI: 10.1037/1528-3542.8.4.494.
- [39] Chenguang Zhang, Jinming Yu, and Zhuang Chen. "Music emotion recognition based on combination of multiple features and neural network". In: *2021 IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. Vol. 4. 2021, pp. 1461–1465. DOI: 10.1109/IMCEC51613.2021.9482244.
- [40] Meixian Zhang et al. "Attention-based Joint Feature Extraction Model For Static Music Emotion Classification". In: *2021 14th International Symposium on Computational Intelligence and Design (ISCID)* (2021), pp. 291–296. DOI: 10.1109/ISCID52796.2021.00074.
- [41] Jiahao Zhao et al. "Multimodal Music Emotion Recognition with Hierarchical Cross-Modal Attention Network". In: *2022 IEEE International Conference on Multimedia and Expo (ICME)*. 2022, pp. 1–6. DOI: 10.1109/ICME52920.2022.9859812.