Analysis of Structural Complexity Features for Music Genre Recognition

Philipp Ginsel
Department of Computer Science
TU Dortmund University
Dortmund, Germany
philipp.ginsel@tu-dortmund.de

Igor Vatolkin, *Member, IEEE*
Department of Computer Science
TU Dortmund University
Dortmund, Germany
0000-0002-9454-9402

Günter Rudolph, *Member, IEEE*
Department of Computer Science
TU Dortmund University
Dortmund, Germany
guenter.rudolph@tu-dortmund.de

***Abstract*—The concept of *structural complexity* describes the temporal progress of feature values on different time scales. We apply it to audio features with the goal to classify music files into genres using k-Nearest Neighbors and Random Forest. We use a publicly available data set of 1550 music tracks which are labeled as belonging to one of six different genres (or to none of them). The classification models are trained with the help of eight feature sets that describe different musical aspects (chords, harmony, instruments, timbre, etc.) in order to find out which features are best suited to predict these genres using the structural complexity. We apply evolutionary multi-objective feature selection to measure individual contributions of different structural complexity features for each genre to feature sets with the smallest classification errors. We also introduce a new feature *chord vector* which is shown to perform significantly better on genre classification with the structural complexity method than the chord features used in a previous work. The statistical analysis of time scales and features leads to several recommendations for the setup of feature processing based on structural complexity.**

***Index Terms*—Music genre recognition, semantic audio features, structural complexity, evolutionary feature selection**

I. Introduction

Structural complexity is a method to measure the temporal change of a base feature on different time scales suggested to correlate to musical complexity as perceived by humans [1]. Other studies introduced similar concepts for the calculation of harmonic complexity to predict genres and styles [2], [3]. Genre classification is a very common task in music information retrieval, because genres are a very popular way to discover music and organize music collections (see survey [4]). Nonetheless, genres are often hard to define, and even for experts it can be difficult to label a song with a single genre.

Common approaches to recognize genres are based on manually engineered audio signal features [5] or deep neural networks in more recent studies [6], [7]. However, classification models built with such descriptors are less interpretable and cannot yield explainable rules why a certain music piece is assigned to a particular genre. A more interpretable solution are fuzzy rule-based systems which were used in [8] together with evolutionary algorithms for music genre recognition. Another approach to this problem is the estimation of structural complexity for semantic audio descriptors, which keeps the feature space as interpretable as possible [9].

In our work, we have applied the k-Nearest Neighbor and the Random Forest classifier together with structural complexity for genre classification and examined the optimal setup of parameters. The results have shown that the choice of a proper time scale depending on the underlying feature group may significantly improve the classification performance. Beyond the estimation of structural complexity for chords, chroma-derived characteristics, harmony, instruments, tempo/rhythm, and timbre, we have implemented a novel feature for chord statistics, which was significantly better than the previous chord complexity vector from [9]. Furthermore, we have applied evolutionary multi-objective feature selection to measure the individual importance of structural complexity feature groups which contribute to feature subsets with smallest classification errors using as few features as possible.

In Section II, we describe the algorithmic backgrounds of structural complexity, audio features, classification methods, and evolutionary feature selection. Section III deals with the setup of experiments. The results are discussed in Section IV. The concluding remarks are given in Section V.

II. Methods

A. Structural Complexity

Structural complexity measures the change of a base feature on different time scales [1]. Let $x_1, x_2, ..., x_S \in \mathbb{R}^M$ be M-dimensional feature vectors that have been extracted for the S frames of a music track. To calculate the structural complexity for the frame $i \in [0, S]$, the W feature vectors before and including x_i are compared to the W feature vectors after x_i. In order to do that, the vectors $x_{i-W}, ..., x_i$ are summarized with a function $s : (\mathbb{R}^M)^W \rightarrow \mathbb{R}^M$. The same is done for the vectors $x_{i+1}, ..., x_{i+W}$. The results are compared with a divergence function $d : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}^+$.

For s and d, we followed [1] using the mean for s and the *Jenson-Shannon* divergence for d. For the time scales, we set $W \in \{2, 4, 8, 16, 32, 64\}$ (measured in seconds). The music tracks were divided in partitions, for which we have calculated the 1st, 2nd, and 3rd quartiles, the minimum, the maximum, the mean, and the variance for each complexity. This resulted in a 7-dimensional vector for each partition for later classification. For $W \in \{2, 4, 8\}$, we used partitions of

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

the size 24s with a 12s overlap; for $W \in \{16, 32\}$, 96s with a 48s overlap, and for $W = 64$, 138s with a 69s overlap.

B. Features

We distinguish between 8 feature sets which describe different musical properties. The structural complexity is calculated on each of these groups separately. We adopt feature sets from [9] extending them with a new feature *chord vector* described below. Short descriptions and examples of the base features are provided in Table I. All features are normalized using *min-max normalization* before the structural complexity is calculated.

For the estimation of the chord vector, we have first extracted the chords with Chordino Vamp Plugin [10]. For structural complexity processing, they have to be converted into numerical values. The feature set *chords* as used in [9] contains numbers of different chords and their changes in 10s. That way, structural complexities can only be calculated for partitions with a length of at least 10s and the values describe some kind of *change of change* because the structural complexity is not calculated on the chords themselves.

For the estimation of the structural complexity on the extracted chords themselves, and not on their change, we convert the chord names into vectors, for which means and distances can be calculated, and the resulting values have some kind of musical meaning. Each chord name is converted into a 20-dimensional binary vector. The first 12 indices of the vector are a one-hot encoding of the root of the chord. Each index represents one note of the chromatic scale. The use of one-hot encoding has the benefit of every index of the vector having its own unique meaning. Enharmonically equivalent chords like D# and Eb have the same representation. The labeling with either D# or Eb is dependent on the context. Sometimes the ambiguity of enharmonic equivalences is even desired by the composer. Therefore, we treat enharmonically equivalent chords in the same way. The indices 13 through 16 are a one-hot encoding of the different types of chords the plugin can detect, i.e. major, minor, diminished, and augmented. The last four indices describe the four different extensions the plugin can detect, i.e. major sevenths, minor sevenths, diminished sevenths, and sixths (although sixths and diminished sevenths are enharmonically equivalent, we separate them, because they have a very different musical meaning). Additional tensions are not represented because they are not extracted by the Chordino Vamp Plugin but can be considered in future work. If the plugin extracts the chord name "N", it means that it either did not recognize the chord or that no harmonic sound was identified in that moment. We represent this with a vector, in which every index is 0.

As a result, we represent every chord as a unique vector with the exception of enharmonic equivalencies and bass notes (D^7 and D^7/F# have the same representation). The reason we used 20 dimensions and not less was that every index of the vector got its own meaning. For example, the first index of the mean of some vectors describes, which ratio of the chords have the root C and the value of the 13th index describes, which ratio of the chords were major chords. That way, the structural complexity of this vector can describe the change of root notes and chord qualities in a given time frame.

C. Classification Algorithms

For classification we used the algorithms *k-Nearest Neighbors* (KNN) and *Random Forest* (RF).

KNN is a very simple classification algorithm. Let $x_1, ..., x_n$ be labeled training instances. Let Y be the set of classes to predict, so that a label $y_i \in Y$ will be assigned to each training instance x_i. To classify an unknown instance x, the algorithm finds the k training instances that have the smallest distances to x according to some metric (we use the Euclidean distance). Then, the most frequent label in these k neighbors is assigned to k. For binary classification, it is best to only use odd values for k, so that it would be impossible to get a tie, cf. [11].

The RF classifier builds an ensemble of small decision trees, each trained to classify the training examples [12]. For each tree, only a subset of all features is randomly selected, in a standard implementation equal to $\sqrt{|X|}$ (X denotes the set of all features). Unknown instances are then classified using majority voting. This method is very robust and usually only the number of trees should be adjusted.

Because we focus on semantic audio features and our data set contained 1550 tracks (see Section III), we did not apply deep neural networks which can be very powerful for classification tasks but require very large training sets and often lead to less interpretable features and models.

D. Feature Selection and Evaluation

To identify the most relevant structural complexity features for individual classification tasks, we have integrated multi-objective evolutionary feature selection (FS) based on $\mathcal{S}$-Metric Selection Evolutionary Multi-Objective Algorithm (SMS-EMOA) [13] as proposed in [14] and adjusted in [15]. The selected features are here represented by a binary vector $\mathbf{q}$ with $\mathbf{q}_i = 1$ indicating that the i-th feature has to be selected and with $\mathbf{q}_i = 0$ to be removed from the training data set.

The first one is the balanced error rate ϵ_r, defined as:

$$\epsilon_r = \frac{1}{2}\left(\frac{FN}{TP+FN} + \frac{FP}{TN+FP}\right), \tag{1}$$

where TP is the number of true positives (classification instances correctly predicted as belonging to the positive category in a binary classification task), TN true negatives (instances correctly predicted as not belonging to the positive class), FP false positives (negative instances predicted as positives), and FN false negatives (positive instances predicted as negatives).

The second optimization criterion is the number of selected features, with the target to produce small and robust feature sets keeping only the most relevant features:

$$f_r = \frac{|\Phi(\mathbf{q})|}{|X|}, \tag{2}$$

where $|\Phi(\mathbf{q})|$ corresponds to the number of selected features and $|X|$ the overall number of features.

TABLE I
DESCRIPTIONS AND EXAMPLES OF THE USED FEATURE SETS

Feature set	Description	Examples of base features
Chords	Chord changes in 10s frames	Number of different chords in 10s
Chord vector	The chord vector feature, see Section II-B	Chord vector
Chroma	Various chroma implementations	Bass chroma [10]
Chroma-derived	Features related to chroma	Chroma maximum
Harmony	Aspects of a track's harmony	Interval strengths estimated from 10 highest semitone values
Instruments	Share of instruments in 10s frames	Share of strings
Tempo and rhythm	Aspects of a track's tempo and rhythm	Estimated beat number per minute
Timbre	Aspects of a track's timbre	Zero-crossing rate

The optimization of SMS-EMOA parameters was beyond the scope of this work, we used the setup which was successful in previous studies. The population size was set to 50 solutions (feature sets) randomly initialized with approximately half of selected features. The number of expected bit flips per generation (mutations) was set to $64/|X|$, with a larger probability to deselect features ($w_1 = 0.05$ and $w_2 = 0.2$ after [15]). The number of generations was 3000 (for the reasons of acceptable convergence behaviour), and we run 10 statistical repetitions for each combination of a classification task and a method. For further details, see [14], [15].

TABLE II
SUMMARY OF PARAMETERS VARIED IN OUR STUDY

Set	Parameter	Values
$\{T\}$	Classification tasks	{Classical, Electronic & Dance, Jazz, Rock & Pop, Rap, R&B & Soul}
$\{G\}$	Feature groups	{Chords, chord vector, chroma, chroma-derived, harmony, instruments, tempo and rhythm, timbre}
$\{W\}$	Window size	{2s, 4s, 8s, 16s, 32s, 64s}
$\{A\}$	Algorithms	{KNN, RF}
$\{K\}$	No. of KNN neighbors	{1, 3, 5, 7, 9, 11, 13, 15}
$\{N\}$	No. of RF trees	{100, 200}
$\{F\}$	Validation folds	{1, 2, ..., 10}

III. EXPERIMENT SETUP

We have selected *1517-Artists* dataset [16] for our experiments, because it is publicly available and contains audio tracks for the extraction of signal-based features. From the original 3180 tracks assigned to 23 music genres, we have selected only tracks from different artists in each category. Only tracks longer than 138s were kept, so that the structural complexity features could be calculated for all window sizes (cf. Section II-A), leading to the final number of 1550 tracks.

Table II lists parameters of our study. We distinguish between six genres for binary prediction in set $\{T\}$. $\{G\}$ contains 8 feature groups described in Section II-B. These features are processed using the structural complexity with six different window sizes, as described in Section II-A. For chords and instruments, only the window sizes $\{W\} = \{16, 32, 64\}$ are used, since these features can only be extracted for 10s windows. $\{A\}$ denotes classification algorithms (cf. Section II-C), $\{K\}$ numbers of neighbors, and $\{N\}$ numbers of trees. $\{F\}$ contains numbers of folds during 10-fold *stratified cross validation* [17]. The music data set is partitioned into 10 folds, such that the share of positive samples is roughly the same in each fold. Because the folds contain much more negative than positive tracks, we evaluate the classification performance with the balanced relative error ϵ_r. All experiments were conducted within the AMUSE-Framework [18].

IV. RESULTS

A. Overview

We show ϵ_r for all experiments with KNN in Fig. 1 and for RF in Fig. 2. Table III lists ϵ_r for the best combinations of k and W with KNN and the best combinations of the number of trees and W with RF. After the separate classification with each complexity feature vector to measure its individual importance, we have applied evolutionary FS (see Section II-D) for the complete feature set with all structural complexities and different analysis frames (294 dimensions). The mean ϵ_r across all folds for the feature set with the smallest error after 10 statistical repetitions for each fold is provided in the last two lines of the table.

Classical was the easiest genre to classify, while chroma complexity was the best feature group. The best results were reached with the combinations Classical and harmony, and Rap and chroma. For Rap, good results were also reached with the feature groups harmony and chord vector. Interestingly, Classical music was classified quite well with instruments, although this group did not work so well with the other genres. We can also see that the new feature chord vector achieved better results than the old chords complexity.

The classification with RF led to smaller errors than with KNN for all genres after feature selection. Also for individual structural complexities RF was often better than KNN, but not always: for instance, recognition of Electronic with chord, chroma, chroma-derived, and harmony complexities (half of all eight groups) had smaller ϵ_r with KNN.

B. Comparison of Structural Complexity Groups in Feature Sets with Smallest Errors

After the application of evolutionary FS as described in Section II-D, we have analyzed feature sets with the smallest errors for 10 folds. These sets were estimated after 10 statistical repetitions of FS and the estimation of non-dominated fronts for final populations. Table IV lists average relative contributions of each structural complexity feature group to these sets (in per cent, the numbers are normalized with regard

TABLE III
ϵ_r OF THE BEST PAIR OF k AND W WITH KNN AND THE BEST PAIR OF THE NUMBER OF TREES AND W WITH RANDOM FOREST

	Alg	Class	Elect	Jazz	Pop	Rap	RnB
Chords	KNN	0.408	0.317	0.430	0.447	0.393	0.454
	RF	0.307	0.318	0.384	0.374	0.473	0.469
Chord vector	KNN	0.357	0.294	0.341	0.422	0.280	0.388
	RF	0.243	0.260	0.259	0.357	0.215	0.363
Chroma	KNN	0.219	0.270	0.366	0.378	0.194	0.371
	RF	0.196	0.290	0.312	0.251	0.176	0.316
Chroma-derived	KNN	0.254	0.361	0.393	0.327	0.327	0.353
	RF	0.207	0.386	0.331	0.286	0.285	0.381
Harmony	KNN	0.177	0.304	0.404	0.424	0.253	0.381
	RF	0.095	0.322	0.388	0.418	0.196	0.386
Instruments	KNN	0.244	0.416	0.350	0.335	0.325	0.414
	RF	0.135	0.409	0.398	0.240	0.238	0.345
Timbre	KNN	0.282	0.300	0.399	0.334	0.307	0.395
	RF	0.191	0.278	0.313	0.236	0.229	0.342
Tempo and rhythm	KNN	0.286	0.299	0.431	0.421	0.350	0.432
	RF	0.237	0.287	0.422	0.377	0.347	0.348
Best	KNN	0.057	0.075	0.145	0.128	0.067	0.174
	RF	**0.054**	**0.070**	**0.136**	**0.100**	**0.057**	**0.141**

TABLE IV
SHARES IN PER CENT OF 8 STRUCTURAL COMPLEXITY FEATURE GROUPS IN SELECTED FEATURE SETS WITH THE SMALLEST ϵ_r

	Alg	Class	Elect	Jazz	Pop	Rap	RnB
Chords	KNN	7.14	4.69	12.11	6.60	12.92	10.40
	RF	3.41	6.11	13.33	7.00	7.45	9.25
Chord vector	KNN	14.97	6.87	18.30	11.67	17.05	19.69
	RF	14.66	10.10	**23.89**	14.87	15.19	10.25
Chroma	KNN	14.61	17.06	**21.22**	16.25	**27.65**	**21.54**
	RF	17.97	14.28	15.82	11.49	**18.66**	**16.68**
Chroma-derived	KNN	5.24	10.67	6.78	**20.98**	6.80	9.61
	RF	11.96	14.92	5.08	16.85	15.84	13.69
Harmony	KNN	**25.68**	17.25	10.56	12.60	12.97	14.10
	RF	**22.37**	14.23	12.68	14.79	9.47	13.31
Instruments	KNN	10.94	3.64	6.33	4.06	2.14	4.09
	RF	7.42	9.26	8.65	6.11	7.30	5.75
Timbre	KNN	12.60	18.81	11.88	19.61	4.96	10.10
	RF	11.81	**15.86**	9.80	**18.68**	13.34	15.47
Tempo and rhythm	KNN	8.82	**21.02**	12.82	8.24	15.49	10.47
	RF	10.40	15.24	10.75	10.20	12.75	15.60

to the number of features in the set with the smallest error, and then averaged across all 10 folds). The numbers in bold mark the largest shares for each category, estimated separately for experiments with KNN and RF.

All structural complexity feature groups contribute to the best selected sets, but, as expected, the importance of feature groups varies across categories. Chords and instruments seem to be the least important groups (for the latter one, possibly because only guitar, piano, string, and wind instrument groups were analyzed in the instrument complexity group [9]). Chroma complexity has the largest share for 5 of 12 cases (6 problems x 2 classifiers), being rather important. Chord vector has higher shares compared to chords for each task and classifier, however, it could not completely replace the chords in the best selected feature sets.

With regard to genres, the identification of classical pieces benefits at most from harmony structural complexity group. For Rap and R'n'B, chroma complexity features have the largest shares. Note that the groups with the largest contribution in Table IV are not necessarily the groups with the smallest errors in Table III. E.g., for Electronic, chroma complexity alone has the smallest $\epsilon_r = 0.270$ with KNN and chord vector with RF ($\epsilon_r = 0.260$), but the groups with largest contributions to the best selected feature sets are tempo/rhythm for KNN and timbre for RF. Generally, we do not recommend to omit the estimation of some "weaker" structural complexity groups, but rather to store all these statistics and to apply feature selection for the identification of the most relevant groups and their combinations for each genre separately. This automatic approach helps to extract musically meaningful semantic information for further theoretical analysis of music genres and styles, or also other categories, like music pieces from a given composer or a particular time decade.

TABLE V
OVERVIEW OF STATISTICAL TESTS

#	Compared	Fixed	Varied	Dim.
1	$\{W_i, W_j\} : i, j \in \{1, ..., \lvert W\rvert\}, i \neq j$	A, G, $\widehat{K}/\widehat{N}$	T, F	60
2a	$\{K_i, K_j\} : i, j \in \{1, ..., \lvert K\rvert\}, i \neq j$	T, A	G, F	80
2b	$\{N_i, N_j\} : i, j \in \{1, ..., \lvert N\rvert\}, i \neq j$	T, A	G, F	80
3	G_1, G_2	$\widehat{W, K}/\widehat{W, N}$	T, F	60

C. Comparison of Analysis Frames

Some of the observations of the previous section can be explained by looking at the structural complexity values that were calculated for different genres and feature groups. Fig. 3 shows the mean structural complexity values of the feature groups and genres and their change depending on the window size. Chroma complexity has a high variance across genres, which explains its distinctive performance. For chord vector and chords, the differences between the genres are much higher for chord vector. That explains its comparatively better performance. Tempo and rhythm performed relatively good for Electronic and Dance. An interesting property of this genre's structural complexity values is that it has very low values for short window sizes but high values for long window sizes. This shows that combining features of different window sizes could be helpful for genre classification.

D. Statistical Tests

For the statistical analysis of results, we compared ϵ_r by means of the Wilcoxon signed rank test. Table V presents an overview of tests. The column "Compared" contains settings to compare, "Fixed" settings, which are the same for both vectors to test, and "Varied" settings, which are varied along the dimensions of vectors to test, and produce a final number of dimensions in column "Dim.".

The goal of the first test was to find the best settings for window sizes during the calculation of structural complexity. For a fixed algorithm from the set $\{A\}$ and a feature group from $\{G\}$, we compare two vectors. The first vector contains errors using window size W_i, the second using W_j. Both vectors have 60 dimensions: for each combination of a classification task and a fold, we store the smallest error across experiments with all values from K resp. N, denoted

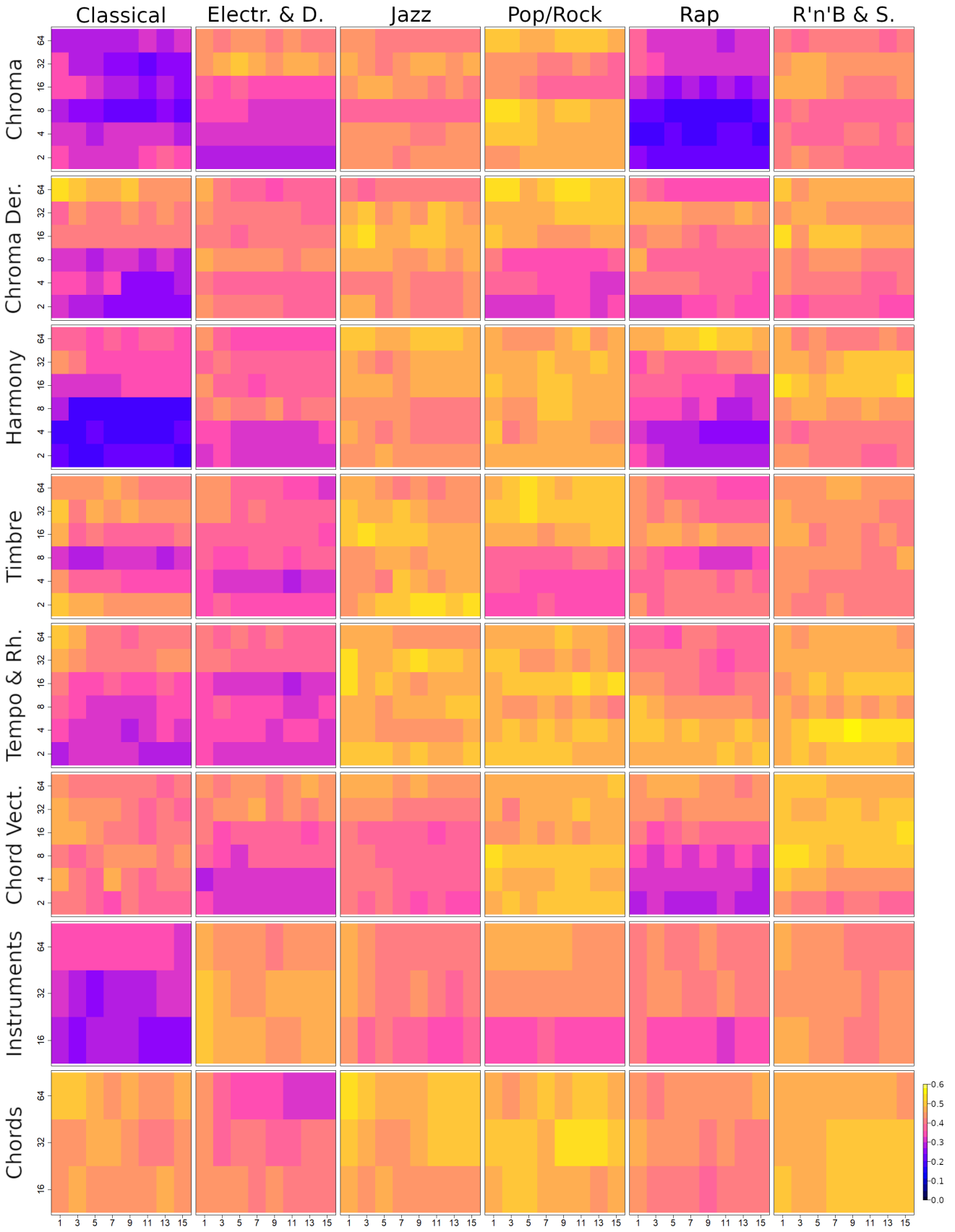

Fig. 1. Balanced relative errors of the tests with the KNN classifier for every combination of feature set, window size, and number of neighbors. The horizontal axis shows the number of neighbors (k). The vertical axis shows the window sizes (W) in seconds. The values of the errors are represented by colors (see legend on the bottom right).

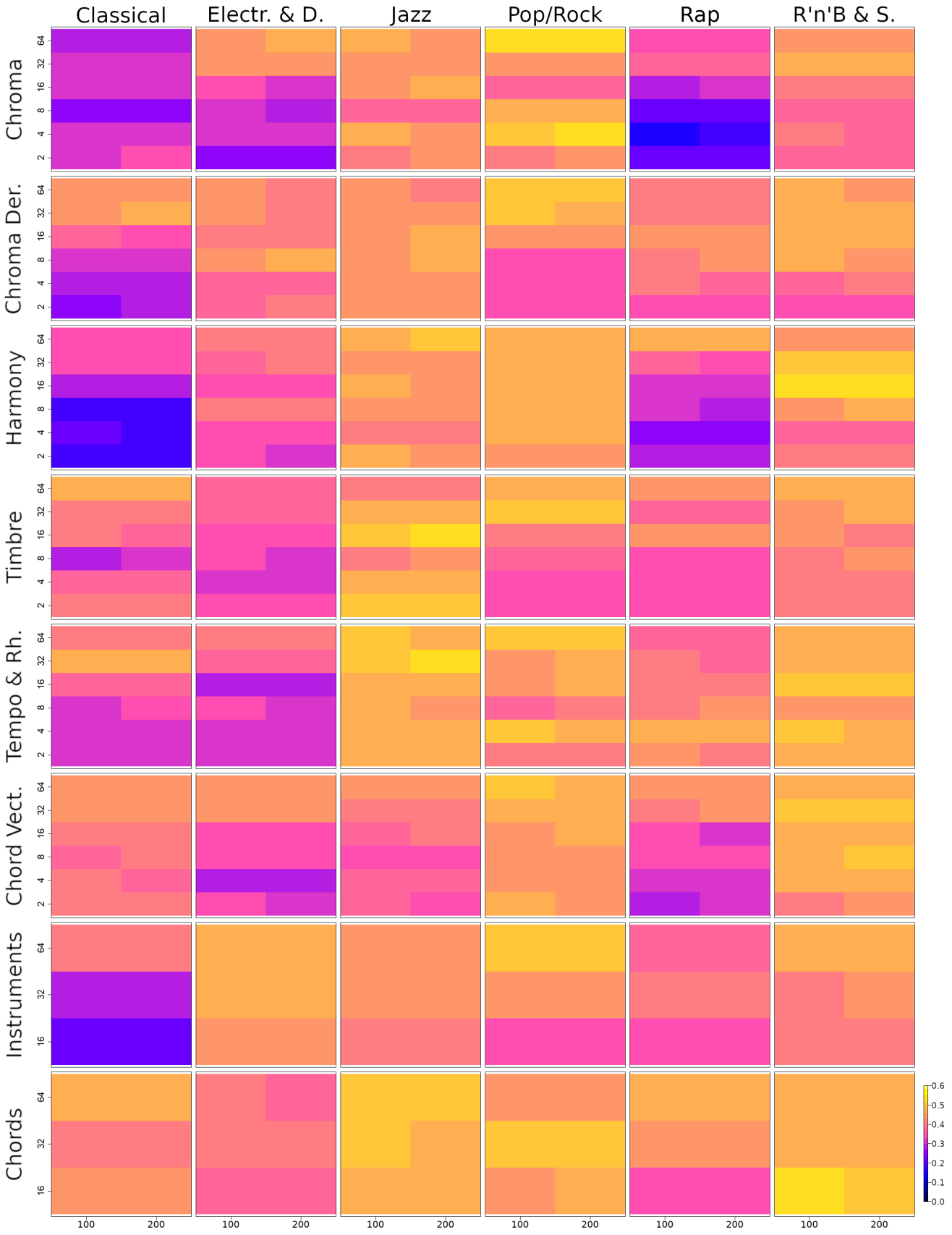

Fig. 2. Balanced relative errors of the tests with the RF classifier for every combination of feature set, window size, and number of trees. The horizontal axis shows the number of trees. The vertical axis shows the window sizes (W) in seconds. The values of the errors are represented by colors (see legend on the bottom right).

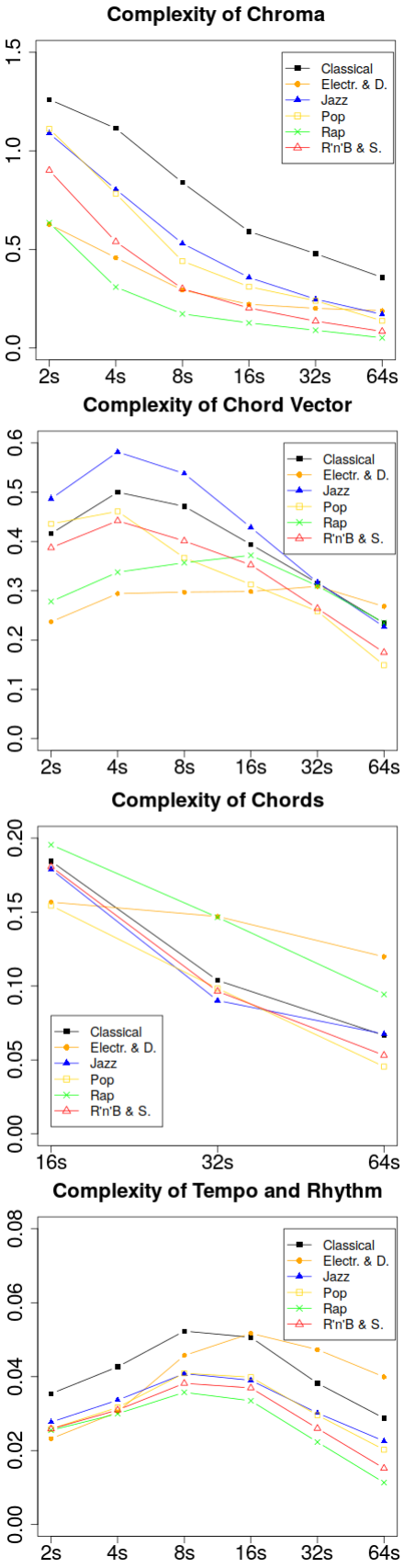

Fig. 3. Mean structural complexity values of the feature groups chroma, chord vector, chords, and tempo & rhythm. The mean complexities are marked on the vertical axes. The window sizes (W) are marked on the horizontal axes.

here with $\widehat{K}$ and $\widehat{N}$. Because all categories contribute to each application of the test at the same time, we can provide here general recommendations for the setup of window size with regard to the feature group.

Table VI shows only significant differences between window sizes. $A \prec B$ denotes that the setting A produced a significantly lower error as B, with p-values listed in columns "p(KNN)" and "p(RF)". For the chroma complexity, the best setting seems to be $W_i = 8$, however, the advantage is more clear for RF compared to KNN. For the chroma-derived complexity, the best setting is $W_i = 2$, however, it is not significantly better than $W_i = 4$. For the chords complexity, no significant differences were estimated. For the chord vector complexity, the best window size is $W_i = 2$, for harmony $W_i = 4$, for instruments $W_i = 16$, and for timbre $W_i = 4$. $W_i = 8$ seems to be quite good for tempo and rhythm.

Generally, smaller window sizes seem to perform better; the window size of 64s could not significantly outperform other sizes for all feature groups. $W_i = 16$ was not better than smaller window sizes (and is recommended for instruments only because it was the smallest possible window size). The choice of $W_i \in \{2, 4, 8\}$ depends on the feature group.

The goal of the second test was to find the best k for KNN and to compare the numbers of trees in RF. For KNN, the best k varies strongly for different classification tasks. For Classical, the only significant observation is that $K_i = 5$ is better than $K_j = 1$ (p=0.02918). For Electronic and Dance as well as for Rap, the best value is $K_i = 11$, which is significantly better than $K_j \in \{1, 7, 9\}$ resp. $K_j \in \{1, 5\}$. For Jazz, the best value is $K_i = 9$, significantly better than $K_j \in \{1, 15\}$. For Pop/Rock and R'n'B/Soul, there were no significant differences across all pairs of k values. The numbers of trees for RF were not significantly different for all classification tasks.

Finally, the third test showed that the proposed chord vector complexity significantly outperformed chords for both algorithms. For KNN, the median error of chord vector across all categories and folds was 0.27818 against 0.33116 (p=2.1885e-07), and for RF, 0.29605 against 0.36525 (p=2.5616e-07).

V. Conclusions

In this work, we tested and optimized feature processing by means of structural complexity in combination with different semantic audio feature groups for music genre classification. With the help of evolutionary multi-objective feature selection, it was possible to identify the best combination of features and to measure the individual contribution of structural complexity groups to feature sets with the smallest classification errors, allowing for a further theoretical analysis of music categories like genres and styles.

We found that the structural complexity method tends to work best for smaller window sizes with the concrete values depending on the base feature group that is used, but also depending on the classification task. Further recommendations for reasonable window sizes were provided with respect to different feature groups. We have also introduced the new

TABLE VI
RESULTS OF THE TEST COMPARING WINDOW SIZES

Comparison	p(KNN)	p(RF)
Chroma		
$W_i = 2 \prec W_j = 32$		0.000179
$W_i = 2 \prec W_j = 64$		1.108e-05
$W_i = 4 \prec W_j = 32$		0.009258
$W_i = 4 \prec W_j = 64$		0.002542
$W_i = 8 \prec W_j = 4$	0.032059	0.023592
$W_i = 8 \prec W_j = 16$		0.016400
$W_i = 8 \prec W_j = 32$	0.004030	2.052e-06
$W_i = 8 \prec W_j = 64$	0.003233	7.827e-07
$W_i = 16 \prec W_j = 32$	0.021114	0.002669
$W_i = 16 \prec W_j = 64$		0.001262
Chroma-derived		
$W_i = 2 \prec W_j = 8$	0.005390	0.009160
$W_i = 2 \prec W_j = 16$	2.550e-06	8.163e-06
$W_i = 2 \prec W_j = 32$	1.036e-05	2.550e-06
$W_i = 2 \prec W_j = 64$	0.000728	2.629e-05
$W_i = 4 \prec W_j = 8$		0.023143
$W_i = 4 \prec W_j = 16$	2.561e-07	7.351e-06
$W_i = 4 \prec W_j = 32$	6.956e-05	1.114e-05
$W_i = 4 \prec W_j = 64$	0.005226	6.816e-05
$W_i = 8 \prec W_j = 16$	8.456e-05	0.026202
$W_i = 8 \prec W_j = 32$	0.001279	0.003101
$W_i = 8 \prec W_j = 64$		0.015127
Chord vector		
$W_i = 2 \prec W_j = 16$	0.006975	0.010161
$W_i = 2 \prec W_j = 32$	0.000563	8.438e-06
$W_i = 2 \prec W_j = 64$	2.160e-05	2.128e-06
$W_i = 4 \prec W_j = 32$	0.031586	3.748e-05
$W_i = 4 \prec W_j = 64$	0.000831	3.404e-05
$W_i = 8 \prec W_j = 32$	0.040344	3.192e-05
$W_i = 8 \prec W_j = 64$	0.008587	0.000671
$W_i = 16 \prec W_j = 32$		0.000619
$W_i = 16 \prec W_j = 64$	0.040805	0.001081
Harmony		
$W_i = 2 \prec W_j = 16$	5.311e-05	0.000540
$W_i = 2 \prec W_j = 32$	9.848e-05	2.534e-05
$W_i = 2 \prec W_j = 64$	0.000540	4.832e-05
$W_i = 4 \prec W_j = 8$	0.041174	0.007372
$W_i = 4 \prec W_j = 16$	1.261e-06	9.528e-05
$W_i = 4 \prec W_j = 32$	8.732e-06	1.057e-06
$W_i = 4 \prec W_j = 64$	1.894e-05	7.353e-06
$W_i = 8 \prec W_j = 16$	0.000342	
$W_i = 8 \prec W_j = 32$	0.000959	0.001199
$W_i = 8 \prec W_j = 64$	0.001435	0.001756
$W_i = 16 \prec W_j = 32$		0.044852
$W_i = 16 \prec W_j = 64$		0.018857
Instruments		
$W_i = 16 \prec W_j = 32$	0.026221	3.042e-05
$W_i = 16 \prec W_j = 64$	0.037897	1.648e-06
Tempo and rhythm		
$W_i = 2 \prec W_j = 4$		0.029718
$W_i = 2 \prec W_j = 32$		0.009978
$W_i = 2 \prec W_j = 64$		0.004091
$W_i = 8 \prec W_j = 4$	0.024747	0.007207
$W_i = 8 \prec W_j = 16$	0.038106	
$W_i = 8 \prec W_j = 32$	0.017072	0.000347
$W_i = 8 \prec W_j = 64$		0.000310
$W_i = 16 \prec W_j = 32$		0.046040
Timbre		
$W_i = 2 \prec W_j = 32$		0.048085
$W_i = 2 \prec W_j = 64$		0.009979
$W_i = 4 \prec W_j = 2$	0.042904	
$W_i = 4 \prec W_j = 16$	0.000578	0.024748
$W_i = 4 \prec W_j = 32$	0.001756	0.001026
$W_i = 4 \prec W_j = 64$	0.019615	0.000446
$W_i = 8 \prec W_j = 2$		0.004227
$W_i = 8 \prec W_j = 16$	0.000888	0.0001216
$W_i = 8 \prec W_j = 32$	0.000347	4.592e-06
$W_i = 8 \prec W_j = 64$	0.007701	1.227e-05

feature chord vector which performed significantly better than related statistics from the previous work.

In future, we plan to test further combinations of structural complexity features (and the base features for each complexity group), together with other processing parameters, like the choice of a distance metric. Further applications include other classification tasks like recognition of emotions or music segmentation.

ACKNOWLEDGMENTS

This work was partly funded by the DFG (German Research Foundation, project 336599081).

REFERENCES

[1] M. Mauch and M. Levy, "Structural change on multiple time scales as a correlate of musical complexity," in *Proc. 12th Int'l Society for Music Information Retrieval Conf. (ISMIR)*, 2011, pp. 489–494.

[2] L. Maršík, J. Pokornyy, and M. Ilcík, "Improving music classification using harmonic complexity," in *Proc. 14th Conf. on Information Technologies - Applications and Theory*, 2014, pp. 13–17.

[3] C. Weiß and M. Müller, "Tonal complexity features for style classification of classical music," in *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, pp. 688–692.

[4] B. L. Sturm, "A survey of evaluation in music genre recognition," in *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation, 10th International Workshop, AMR*, 2012, pp. 29–66.

[5] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[6] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proc. IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2392–2396.

[7] Y. M. Costa, L. S. Oliveira, and C. N. Silla, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied Soft Computing*, vol. 52, no. C, p. 2838, 2017.

[8] F. Vega, F. Chávez, R. Alcala, and F. Herrera, "Musical genre classification by means of fuzzy rule-based systems: A preliminary approach," in *IEEE Congress of Evolutionary Comp. (CEC)*, 2011, pp. 2571 – 2577.

[9] I. Vatolkin, "Improving supervised music classification by means of multi-objective evolutionary feature selection," Ph.D. dissertation, Technische Universität Dortmund, Dortmund, 2013.

[10] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. 11th Int'l Society for Music Information Retrieval Conf. (ISMIR)*, 2010, pp. 135–140.

[11] J. M. Keller, M. R. Gray, and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 4, pp. 580–585, July 1985.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[14] I. Vatolkin, M. Preuß, and G. Rudolph, "Multi-objective feature selection in music genre and style recognition tasks," in *Proc. 13th Annual Genetic and Evolutionary Computation Conf. (GECCO)*, 2011, pp. 411–418.

[15] I. Vatolkin and D. Stoller, "Evolutionary multi-objective training set selection of data instances and augmentations for vocal detection," in *Proc. 8th Int'l Conf. on Computational Intelligence in Music, Sound, Art and Design (EvoMUSART)*, 2019, pp. 201–216.

[16] K. Seyerlehner, G. Widmer, and T. Pohle, "Fusing block-level features for music similarity estimation," in *Proc. 13th Int'l Conf. on Digital Audio Effects (DAFx)*, 2010, pp. 225–232.

[17] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 1995, pp. 1137–1143.

[18] I. Vatolkin, W. M. Theimer, and M. Botteck, "AMUSE (advanced music explorer) - A multitool framework for music data analysis," in *Proc. 11th Int'l Society for Music Information Retrieval Conf. (ISMIR)*, 2010, pp. 33–38.