Audio Engineering Society

Convention Paper 9997

Presented at the 144th Convention
2018 May 23 – 26, Milan, Italy

Feature Selection for Dynamic Range Compressor Parameter Estimation

Di Sheng[1] and György Fazekas[1]

[1] *Centre for Digital Music (C4DM), Queen Mary University of London*

Correspondence should be addressed to Di Sheng (d.sheng@gmail.com)

ABSTRACT

Casual users of audio effects may lack practical experience or knowledge of their low-level signal processing parameters. An intelligent control tool that allows using sound examples to control effects would strongly benefit these users. In previous work, we proposed a control method for the dynamic range compressor (DRC) using a random forest regression model. It maps audio features extracted from a reference sound to DRC parameter values, such that the processed signal resembles the reference. The key to good performance in this system is the relevance and effectiveness of audio features. This paper focusses on a thorough exposition and assessment of the features, as well as the comparison of different strategies to find the optimal feature set for DRC parameter estimation, using automatic feature selection methods. This enables us to draw conclusions about which features are relevant to core DRC parameters. Our results show that conventional time and frequency domain features well known from the literature are sufficient to estimate the DRC's threshold and ratio parameters, while more specialised features are needed for attack and release time, which induce more subtle changes to the signal.

1 Introduction

The area of intelligent audio production has been developing fast over the last decade. Common paradigms adopted by researchers in this field include the extraction of low-level audio features to control important aspects of audio processing directly, as in adaptive effects [1], or the use of machine learning models to associate audio features with higher-level semantic concepts that may be used to describe sounds. Notable previous works include the use of semantic terms to control audio effects as discussed in [2], as well as the more recent SAFE system [3] which allows the association and recall of effect settings with semantic descriptors such as *warm* or *bright*. These systems rely on crowd-sourced data. Machine learning approaches to learn from expert parameter choices are discussed in [4], while some recent works use NMF [5] and deep learning [6] to learn from the difference between the pre- and post-processed audio. Researchers also developed new interfaces to enhance the workflow in the production process [7]. A thorough review of intelligent production with a particular focus on automatic or semi-automatic mixing is provided in [8].

Our research is different from most previous works in that it aims to estimate the parameters of the DRC given a sound example, such that the processed audio sounds similar in some relevant perceptual attributes (e.g. timbre or dynamics) to the reference sound. A thorough review of DRC's design and analysis is provided in [9]. The initial intelligent system design proposed in [10] has the following components: 1) an audio feature extractor that generates features corresponding to

each parameter, 2) a regression model that maps audio features to audio effect parameters and 3) an audio similarity measure to compare the processed and the reference audio. This research focuses on the improvement of the first component, namely, feature extraction, which is key to achieve good performance using the regression model within this context.

Conventional audio features as well as feature designed specifically to estimate DRC parameters may have redundancies amongst them. This paper aims to find the optimal feature set using feature selection methods. We consider several selection strategies to balance relevance, generality, and performance. The rest of the paper is organised as follows: Section 2 describes the relevant background and related works in feature selection. The feature set is provided in Section 3, followed by a discussion of the feature selection strategies we used in Section 4 and finally our evaluation and results in Section 5. Conclusions and future works are outlined in Section 6.

2 Background

Feature selection is a commonly used data preprocessing technique. By selecting the optimal features using large amounts of data, we can reduce the number of features, remove irrelevant or redundant features, reduce computational cost and better deal with noisy data. This brings immediate benefits for applications: speeding up data mining algorithms and improve mining performance such as predictive accuracy and result comprehensibility [11][12]. The general steps of feature selection are subset generation, subset evaluation, selecting a stopping criterion and result validation. When classifying the methods by subset generation, the strategies can be put into these categories: 1) complete search, 2) sequential search and 3) random search. At subset evaluation stage, feature selection algorithms can be classified as *filter* model, *wrapper* model, and *hybrid* model. Stopping criterion is normally defined as the search being complete, or reaching a certain given bound such as the number of iterations. The most common result validation method is the direct use of the machine learning model performance. In this research, we will use complete search and will alter two subset evaluation strategies: the filter model and the wrapper model. The former considers the relations across features while the later considers the algorithm performance. In terms of audio feature applications, authors in [13] considered several correlation based filter models, while in [14], researchers applied a wrapper model. There are embedded methods which combine selection strategy with machine learning algorithms. Since random forest regression is used in this research, we will also consider several feature ranking algorithms specific to random forest. Originally proposed in [15] and implemented in [16][17], the feature importance method measures the change in the out-of-bag (OOB) error rate for each individual tree when replacing a certain feature with random values. The average performance change can be used as a measure of feature significance. In addition, another commonly used method described in [18] can be referred to as *mean decrease impurity*. This is defined as the total decrease in node impurity averaged over all trees of an ensemble. The node impurity can be approximated by the proportion of samples reaching a certain node.

3 Feature Extraction

This section provides details about the feature set we used and developed. In our research, we focus on four DRC parameters, threshold, ratio, attack and release time. These are the main parameters of the DRC. Other parameters e.g. the make-up gain are relatively simple to predict. We use standard frequency and time domain features as the universal features for the prediction of all parameters, as described in subsection *I, II*. We design features specifically for ratio, attack and release time, as detailed in subsection *III*. The basic signal processing algorithms are implemented using Essentia as well as the Numpy and Scipy packages in Python.

Similarly to [10], in this work we focus on features applicable to isolated notes with a view to extend our approach to audio loops and more complex polyphonic material using audio decomposition techniques such as NMF in future work.

I. Frequency Domain Features

Frequency domain statistical features are the most commonly used features for predicting DRC parameters [4][19] because the statistical features are heavily related to dynamics. In this research, we define the magnitude spectrogram $Y(n,k) = |X(n,k)|$ with $n \in [0 : N-1]$ and $k \in [0 : K]$ where N is the number of frames and k is the frequency index of the STFT, $X(n,k)$, of the input audio signal with a window length of $M = 2(K+1)$. Since the signal $Y(n,k)$ has two dimensions, we use two typefaces to distinguish the mean and variance over the time dimension and the

frequency dimension. $\mathfrak{E}$ and $\mathfrak{Var}$ are used to represent the calculation over time, while E and Var are used across frequency. The mean and variance operations across both dimensions are aimed to capture most of the dynamic changes. The features related to the first order statistical frequency feature, spectral centroid, are given in Eqn.1-2. We extend the list until the 4th moment, therefore, the frequency feature list is as follows: SC_{mean}, SC_{var}, SV_{mean}, SV_{var}, SS_{mean}, SS_{var}, SK_{mean}, SK_{var}, where SC stands for spectral centroid, SV stands for spectral variance, SS for spectral skewness, and SK for spectral kurtosis.

$$\mathsf{SC}_{\mathsf{mean}} = \mathfrak{E}\left(\frac{\sum_{k=0}^{K-1} k * Y(n,k)}{\sum_{k=0}^{K-1} Y(n,k)}\right), \tag{1}$$

$$\mathsf{SC}_{\mathsf{var}} = \mathfrak{Var}\left(\frac{\sum_{k=0}^{K-1} k * Y(n,k)}{\sum_{k=0}^{K-1} Y(n,k)}\right), \tag{2}$$

Additionally we extract MFCC features. As the Cepstrum represents the envelope of Mel-scaled spectrograms, MFCC are commonly used to represent certain aspects of the timbre of an audio signal. Given frame-wise MFCCs, $M(n,k)$, with $k \in [0,13]$ represents the first 13 Mel-frequency Cepstrum coefficients, and $n \in [0 : N-1]$ represents the index of the time frame. Using $M(n,k)$ to replace $Y(n,k)$ in Eqn.1-2, we can obtain statistical feature based on MFCCs. The higher order statistical frequency characteristics are included in the previous frequency domain features, therefore, we only use the mean and variance of the first two moments of MFCCs, i.e. MC_{mean}, MC_{var}, MV_{mean}, MV_{var}. We believe these features are able to capture necessary statistical characteristics, therefore, the delta and higher-order delta MFCC features are not used for this research.

II. Temporal Features

We calculate statistical features in time domain as well in the same fashion as the frequency domain features. Unlike spectrograms, time domain audio samples are in one dimension. Therefore, we calculate the mean and variance up to the second moment of $x(m)$, the magnitude of audio sample m within each M-length frame. We therefore have $T1_{mean}$, $T1_{var}$, $T2_{mean}$, $T2_{var}$ as time domain features.

RMS features are considered as well using the RMS curves also with a window size of M. The mean and variance, RMS_{mean} and RMS_{var} across N time frames which correspond to the average and variance of energy are also used as a temporal features.

III. Features specific to DRC parameters

Although parameters are not working independently in the DRC process, it is still possible to design specific features that reflect the role of each parameter. In this section, we introduce one feature for ratio and six features for attack and release time respectively.

The feature for ratio is averaging all the sample amplitude above the threshold, assuming we have already predicted a fairly accurate threshold before the prediction of ratio. The energy reflects the ratio directly, except for the attack and release phases, where a smooth curve instead of the real ratio is applied.

$$\mathsf{R}_{\mathsf{a}} = \frac{1}{M}\sum_{m=0}^{M-1} |x(m)|, \forall |x(m)| > threshold \tag{3}$$

Since the attack and release times are parameters that affect only a certain phase of the audio, we design attack/release phase related features to improve the prediction. Eqn.4-6 are the features representing the length, the average energy of the attack phase, and the energy at the end of the attack phase, where the attack time T_A is calculated using the RMS envelope through a fixed thresholding method (cf. [20]). The end of the attack, N_{endA}, is considered to be the first peak that exceed 90% of the maximum RMS energy and the start of the attack, N_{startA}, is the first sample of the RMS envelope that exceed 10%. The RMS curve is smoothed by a low-pass filter with a normalised cut-off frequency of 0.47 rad/s.

$$\mathsf{T}_{\mathsf{A}} = (N_{endA} - N_{startA})/Fs, \tag{4}$$

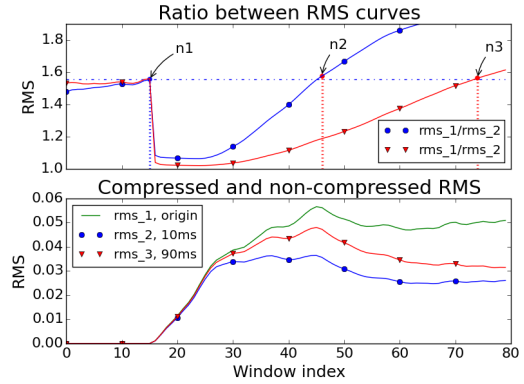
$$\mathsf{A1}_{\mathsf{att}} = \frac{1}{N_{endA} - N_{startA}} \sum_{n=N_{startA}}^{N_{endA}} rms\_curve(n), \tag{5}$$

$$\mathsf{A2}_{\mathsf{att}} = rms\_curve(N_{endA}), \tag{6}$$

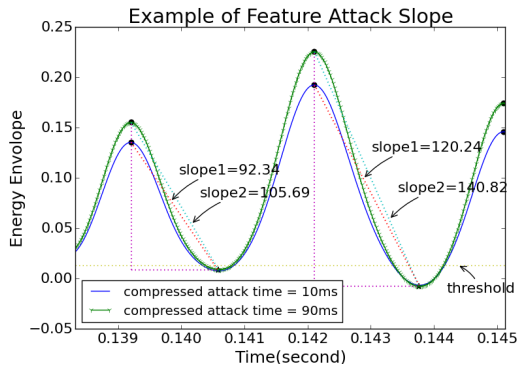
Procedure 1 Calculate $A3_{att}$

Input:
rms1 : non-compressed audio rms curve;
rms2 : compressed audio rms curve;
Output:
$A3_{att}$;
1: γ = rms1/rms2
2: $n1 \rightarrow \forall\gamma[0:n1] \leq 1.0$
3: $n2 \rightarrow \forall\gamma[n1:n2] \leq 1.0$
4: $A3_{att} = n2 - n1$

Additionally $A3_{att}$ is calculated using the pseudocode shown in Procedure 1. For visualisation, we plot an example in Fig.1(a). The ratio between the time-varying amplitudes of input or original sound and the reference sound is shown in the upper figure. The ratio curve before intersect "n1" is the noise part before the actual audio content. The noise passed through a system will generate an arbitrary gain, so we can find the start of the audio through this ratio curve (intersect "n1") while it can also be used as a threshold to find when the ratio rise back to the threshold of interest (intersect "n2" and "n3"). The distance between the two dots clearly shows the speed of the operation of compressor, where a short distance between "n2" and "n1" is corresponding to a small attack time, and the longer distance between intersect "n3" and intersect "n1" is for a longer attack time.



(a) Feature $A3_{att}$



(b) Feature $A4_{att}$, $A5_{att}$

Fig. 1: Examples to demonstrate the procedure of generating attack time features

$A4_{att}$,$A5_{att}$ are calculated using the pseudocode in Procedure 2. During the transient part of the note, there are most likely several ripples. The slope of the ripples can reflect how fast the compressor operates, c.f. Fig.1(b). The RMS curve which has larger energy slope responds to a longer attack time. Based on this observation, we calculate $A4_{att}$,$A5_{att}$, which corresponding to the mean and variance of the slopes. Features corresponding to release time, T_R, $A1_{rel}$-$A5_{rel}$, are calculated in the same fashion at the release phase.

Procedure 2 Calculate $A4_{att}$, $A5_{att}$;

Input:

rms : the rms curve of the input audio;
threshold : the threshold of the compressor;

Output:

$A4_{att}$, $A5_{att}$.

1: $\text{rms1} \leftarrow \text{rms}[N_{startA} : N_{endA}]$
2: $\omega \in \Omega$, $\Omega =$ *peaks over threshold in sequence rms1,*
3: $\phi \in \Phi$, $\Phi =$ *notches after each peak,*
4: **for** $\omega_i \in \Omega$ **do**
5: $s_i = (\omega_i - \phi_i)/dist(\omega_i - \phi_i)$
6: **end for**
7: $A4_{att} = \frac{1}{M}\sum_{i=0}^{M} s_i$
8: $A5_{att} = [\frac{1}{M}\sum_{j=0}^{M}(s_j - \frac{1}{M}\sum_{i=0}^{M} s_i)^2]^{1/2}$

Overall, we generated 25 features for each note. Within this set, 18 are for threshold, 19 are for ratio and 24 for attack and release time. There is a possibility that the features contain redundancy. Especially the frequency domain features may not be able to reflect the change of attack and release times significantly, albeit we may assume that these temporal processes affect the spectral characteristics and/or perceived timbre of sounds. Therefore, we assess feature relevance with respect to individual parameters using different feature selection processes described in the next section.

4 Feature selection

The two typical categories of feature selection are model dependent methods and model independent methods. Others may refer to them as "wrapper model" and "filter model". Wrapper models rank feature sets by measuring system performance. This can be fitted into any machine learning framework, however, it may suffer from over-fitting [21]. Filter methods rank features by measuring relevance between the feature and a label or across features. The measurement may be correlation or mutual information. Filter models normally have less computational cost compared to

wrapper models, however, they do not consider performance. This drawback may lead to suboptimal selection of features from the perspective of the learning task. We report the details of the models along with the selection results of the 24 features for attack time in Section 4.1, 4.2, 4.3. The audio set is generated by manually compressing 60 violin notes from RWC isolated note database [22] using 100 attack time settings within the range of (0,100]ms using the DRC designed in SAFE project [23]. The features are extracted from the audio above. Section 4.3 illustrates two strategies we employed specifically for random forest regression. The same procedures are applied to the rest of the three parameters and the final results are reported in Section 5.1.

4.1 Filter Model

I. Ranking features based on the relevance between the features and the label

The first and simplest strategy of feature ranking using the filter model is based on the relevance between the label and the feature, where the labels are the parameter values used as training target for the regression model. We calculate Pearson correlation coefficient [24] and adjusted mutual information [25] as the two measurements for relevance. The higher the correlation or mutual information is, the more important the feature is. The ranking results are given in Table 1 for the attack time feature set. We use different superscript to represent three types of features. * is for frequency domain features, † for designed features, and the rest are temporal features.

Corr	$RMS_{mean} > T1_{mean} > MV^\star_{var} > T2_{mean} > MC^\star_{var} > RMS_{var} > T1_{var} > A3^\dagger_{att} > T2_{var} > A4^\dagger_{att} > A5^\dagger_{att} > A2^\dagger_{att} > A1^\dagger_{att} > T^\dagger_A > SK^\star_{var} > SC^\star_{var} > SK^\star_{mean} > SK^\star_{mean} > SV^\star_{var} > SS^\star_{mean} > SV^\star_{mean} > SC^\star_{mean} > MV^\star_{mean} > MC^\star_{mean}$
Mu_info	$MV^\star_{mean} > MV^\star_{var} > MC^\star_{mean} > RMS_{mean} > T1_{mean} > T2_{mean} > MC^\star_{var} > T1_{var} > RMS_{var} > A3^\dagger_{att} > T2_{var} > A2^\dagger_{att} > A4^\dagger_{att} > A5^\dagger_{att} > A1^\dagger_{att} > T^\dagger_A > SC^\star_{mean} > SV^\star_{mean} > SC^\star_{var} > SV^\star_{var} > SS^\star_{mean} > SK^\star_{mean} > SK^\star_{var} > SV^\star_{var}$

Table 1: Ranking for attack time features based on two relevant measure, *Corr* for cross-correlation, and *Mu_info* for mutual information.

Both methods choose temporal features in a higher ranking position than frequency domain features, except for the MFCC features in the mutual information case. It partially proved our assumption that the frequency domain features can not give much information for attack and release time prediction. This theory is investigated further in subsequent feature selection experiments using different methods.

II. Ranking features based on the relevance across the features

The previous ranking method is able to show the relevance between features and the label. However, it does not exploit redundant information between features or discard features that contain overlapping information. It is possible that two features are highly related and actually only one is needed. For this reason we examine relevance across all features.

Fig.2 shows a dendrogram resulting from clustering features using mutual information. For the purpose of demonstration, we plot $1 - mutual\_info$. The result seems reasonable since it groups temporal features together. We observe the same effect for frequency domain features as well. The rule here is to choose the features such that redundant information is reduced. If two features have a high mutual information, we use only one of them instead of both. Based on this rule, we set a threshold and select all features within the clusters which have the mutual information above the threshold. For the clusters lower than the threshold, we choose one feature using the Max-Relevance and Min-Redundancy (mRMR) strategy [26]. The condition is described in Eqn.7, where X represents the full feature set, and S_m is the m-sized cluster where one feature needs to be selected. The condition maximises the mutual information between the feature and label while minimised the mutual information for the selected feature and all the features outside of the selected cluster. In this experiment, we experimentally set the threshold to 0.5.

$$\max_{x_j \in S_m} [I(x_j;c) - \frac{1}{m} \sum_{x_i \in X - S_m} I(x_j;x_i)] \tag{7}$$

The resulting selected features are as follows:

$MC_{mean}, SC_{var}, SK_{mean}, SV_{mean}, T_A, A5_{att}, A3_{att}, MC_{var}$;

Since this method compares relevance across features, the clustering tends to put the same type of features together, e.g. frequency domain features are grouped together. Theoretically, we discard the repeated features in terms of mutual information. However, the

process does not consider if the features are related to the problem. This method will yield features that provide the most information, but not necessarily the ones most related to the target label.

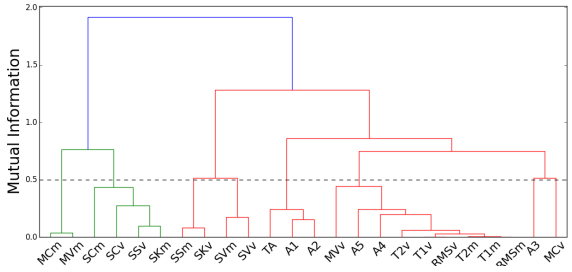


Fig. 2: Relevance between features

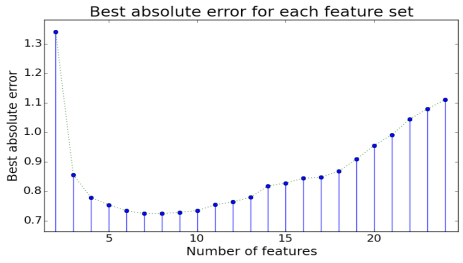


Fig. 3: Best accuracy

4.2 Wrapper model

At this stage, we apply the wrapper model of feature selection suggested in [14], and originally introduced in [12]. The selection strategy is described using the pseudo-code in Procedure 3. This algorithm avoids exhaustive search and hence reduces the computation time significantly. N in the pseudo-code represents the feature set size set to 24. The start point of the algorithm n is set to 2. For each iteration, top m feature sets are passed to the next step, we set $m = 6$. The algorithm stops when the sub_set features size equals to the full_feature size. The best_features are sorted according to the regression performance. We use repeated random sub-sampling validation (Monte Carlo variation) for evaluation, such that the dataset is split into 90% training and 10% testing. The process is repeated 100 times and the average of the Mean Absolute Error (MAE) is used as the performance measure. The best prediction accuracy is displayed in Figure 3. The best performance is provided when we use 7-10 features, since the value is very close in this range. We tend to choose less features in wrapper models to reduce overfitting. In this case we choose 8 features:

$T_A, A1_{att}, T1_{mean}, T2_{mean}, MC_{mean}, MC_{var}, MV_{mean}, MV_{var}$.

The wrapper model is able to provide the best possible accuracy, but it may be overfitted to this particular dataset and therefore may loose its universality or generalisability.

Procedure 3 Feature selection using wrapper model

```
full_set = D(F_0, F_1, ..., F_{N-1})
sub_set = combination(N, n)
for i ∈ [n+1 : N] do
  best_feature = sort(evaluation(∀{sub_set}))
  sub_set = best_feature[0:m]
  for j ∈ [1 : m] do
    for k ∈ [1 : N] do
      if full_set[k] ∉ sub_set[j] then
        sub_set[j].append(sub_set[j], full_set[k])
      end if
    end for
  end for
end for
```

4.3 Feature significance

We introduced two methods specifically designed for random forest. The first method randomises the value of a certain feature and use the change in the out-of-bag (OOB) error rate to assess feature significance. Assuming we have feature set $\boldsymbol{X} = \{X_0, ...X_j, ...X_M\}$, and the task is to rank the M features. We grow T decision trees. For each decision tree t, we consider calculating the prediction error using the out-of-bag samples OOB_t as $\varepsilon = errOOB_t$. If replacing X_j to random values, it will lead to a new error $\widetilde{\varepsilon}_t$. The variable importance is defined as $VI(X_j) = 1/T \sum_t^T (\widetilde{\varepsilon}_t - \varepsilon_t)$. In our implementation, the feature set size M is set to 24, and we grow $T = 10$ decision trees. The top 10 most significant features are selected as follows:

$MC_{mean}, MC_{var}, MV_{mean}, MV_{var}, RMS_{var}, A4_{att}, SV_{mean}, RMS_{mean}, A5_{att}, A2_{att}$;

The second approach uses the decrease in node impurity to decide on the feature importance. For this method, the amount of trees T need to be larger than the number of features M. Therefore we choose $T = 100$. The most important features are chosen as follows:

$A3_{att}, MC_{mean}, MC_{var}, SC_{var}, SS_{mean}, A2_{att}, A5_{att}, A1_{att}, MV_{var}, MV_{mean}$;

5 Results and Evaluation

In this section, we report the final selection results for all four parameters as well as the analysis of the overall performance. The evaluation across each selection algorithms will be presented, along with the relations across each parameter. The feature set is generated using 60 violin notes from the RWC isolated note database [22]. Feature sets for threshold, ratio, release time are generated in the same fashion as the attack time, where we manually set 100 settings for each parameter within (0,50]dB for threshold, (0,20] for ratio, (0,1000]ms for release time using the SAFE DRC. For the random forest regression model, the feature sets are the training data while the training targets are the parameter values.

5.1 Overall performance

Six selection algorithms are demonstrated in Section 4, where we use the features for attack time as an example. Based on the selection results from 6 algorithms, we consider the wrapper model in advance of other models, since it will guarantee optimal performance. The balancing strategy is to choose the wrapper model results plus the features that are selected more than 4 times among the 5 algorithms. The selected feature set is given in Table 2 for all four parameters.

Parameters	Selected Features
Threshold	$MC_{mean}, MC_{var}, MV_{mean}, RMC_{mean}, SC_{mean}$, $SC_{var}, SV_{mean}, SV_{var}, SS_{var}, SK_{mean}, SK_{var}$.
Ratio	$MC_{var}, MV_{var}, T1_{mean}, RMC_{mean}, RMC_{var}$, $SC_{mean}, SC_{var}, SV_{mean}, SV_{var}, SS_{mean}, SS_{var}$, SK_{var}, R_a.
Attack time	$T1_{var}, T2_{var}, A3_{att}, A1_{att}, T_A, A2_{att}$, $A5_{att}, MC_{mean}, MC_{var}, MV_{mean}, MV_{var}$.
Release time	$T_R, A1_{rel}, A2_{rel}, A4_{rel}, A5_{rel}, T1_{mean}, T1_{var}$, $RMS_{mean}, RMS_{var}, MC_{mean}, MV_{var}, SV_{mean}$.

Table 2: The final selected features for four parameters, balancing the selection models.

The selection results for threshold and ratio show that the most related features for these are frequency domain features and MFCC features, while for attack time and release time, the frequency domain features are the least frequently selected. MFCC features are the most popular among all four features, due to their relation with frequency envelope as well as timbre. The prediction error comparison across models are provided in Table 3, where the values are the average of MAE calculated through Monte Carlo variation. Here, we randomly split 10% of the dataset as testing data 100 times and report the average. Variances for each case are very small, about 0.006 for threshold, 0.01 for ratio, 0.007 for attack time and 0.7 for release time in average showing stable performance.

Parameters	Threshold	Ratio	Attack	Release
Selected	**1.242dB**	**0.919**	**0.830ms**	**9.265ms**
Full-list	1.295dB	0.950	1.122ms	12.572ms
Corr	1.808dB	1.103	0.978ms	12.259ms
Mu_info	1.461dB	0.934	0.837ms	12.157ms
Across	1.663dB	1.016	1.147ms	11.635ms
RF_1	1.452dB	0.987	0.908ms	12.604ms
RF_2	1.580dB	1.098	0.982ms	13.808ms
Wrapper	**1.218dB**	**0.892**	**0.725ms**	**8.759ms**

Table 3: Prediction MAE comparing the selected features, full set, and individual selection results.

The final selected feature sets balanced all selection results. The performances are comparable with the best performance selected by the wrapper model, and better than the filter models, random forest feature importance methods, and the full feature set. The results indicate that the selection improves the error rate, and the selected feature sets are much smaller than the full feature set, which also reduces the computational cost.

The selection results of each algorithm and each parameter is represented in Fig.4-7. The results for threshold and ratio show a preference for frequency domain statistical features. The most commonly selected features in case of threshold are $SC_{mean}, SV_{mean}, SK_{mean}, MC_{mean}$, and SS_{mean} for ratio. The most commonly selected feature for attack time is MC_{var} which has been selected by all methods. For release time, it is MV_{var} which is an MFCC derived feature as well. The features designed specific for the attach/release phase are also selected frequently for these two parameters. Fig.6 shows a clear trend that all methods overlook frequency domain features, which fits the assumption that conventional features from literature are not the best choices when predicting these parameters. Fig.7 does not show exactly the same trend as Fig.6, however, the wrapper model does not choose any frequency domain feature, which means even with a certain relevance, frequency domain features may harm the performance (c.f. Fig.3, after the optimised feature set, adding more features increases the error rate). Therefore, we can state that conventional features are satisfactory to predict threshold and ratio, but to predict attack time and release time, the specific designed features are necessary.

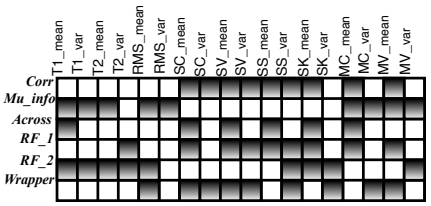


Fig. 4: Selection results of 6 algorithms for threshold

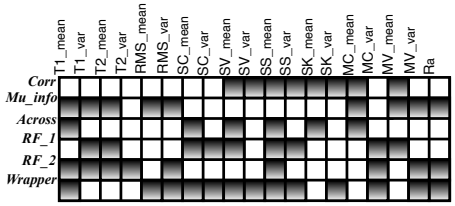


Fig. 5: Selection results of 6 algorithms for ratio

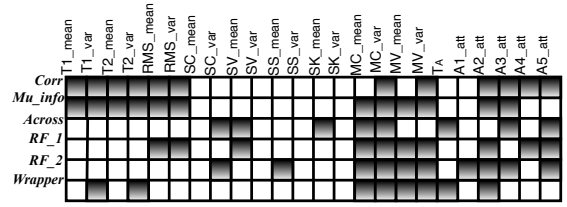


Fig. 6: Selection results of 6 algorithms for attack time

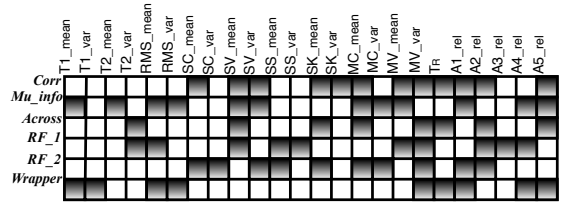


Fig. 7: Selection results of 6 algorithms for release time

5.2 Relations across features

The observation in the overall performance is that attack and release time tend to select similar features, while threshold and ratio likewise exhibit similar behaviour. The overlap rate of the selected features across two pairs and six algorithms are displayed in Table 4. Number *I* to *VI* represent $Corr, Mu\_info, Across, RF\_1, RF\_2$, and $Wrapper$ as in Table 3 respectively. Except for the correlation selection result for attack and release time, all overlap rates are higher than 50%. The results fit the assumption that threshold and ratio have their similarity since they are more directly affecting dynamic range. Attack time and release time are also similar due to the fact that they are both timbre related parameters and they both related to the speed of DRC's action.

Overlap	*I*	*II*	*III*	*IV*	*V*	*VI*
Attack/Release	0.17	0.58	0.50	0.60	0.70	0.56
Threshold/Ratio	0.89	0.89	1.00	0.56	0.56	0.69

Table 4: Overlap rate for parameter pairs

5.3 Relations across algorithms

In this section, we analyse the selected features overlap rate across each selection algorithm for each parameter. Since different algorithms do not guarantee the selection of the same amount of features, the overlap tables do not represent diagonal matrices. The overlap rate for row i and column j is calculated as follows: $rate = \#overlap(i,j)/\#feature(i)$. Table 5-8 represent the overlap rates for 4 parameter features.

Overlap	*I*	*II*	*III*	*IV*	*V*	*VI*
I	1	0.22	0.56	**0.89**	0.22	0.67
II	0.22	1	0.22	0.22	**0.56**	0.33
III	**0.83**	0.33	1	**0.83**	0.17	0.33
IV	**0.89**	0.22	0.56	1	0.33	0.56
V	0.22	**0.56**	0.11	0.33	1	0.33
VI	**0.67**	0.33	0.22	0.56	0.33	1

Table 5: Overlap rate of 6 algorithms for threshold features

Overlap	*I*	*II*	*III*	*IV*	*V*	*VI*
I	1	0.22	0.56	0.56	0.11	**0.67**
II	0.22	1	0.33	0.33	**0.67**	0.56
III	**0.83**	0.50	1	0.50	0.33	0.50
IV	0.56	0.33	0.33	1	0.44	**0.67**
V	0.11	**0.67**	0.22	0.44	1	**0.67**
VI	**0.46**	0.38	0.23	**0.46**	**0.46**	1

Table 6: Overlap rate of 6 algorithms for ratio features

Comparing Table 5-8, one of the common trend is that the wrapper model has the highest overlap with the filter models using correlation and mutual information. It indicates that the features that are able to produce the optimal result are the ones that have the strongest relevance with the label. However, the two types of filter models do not have a high overlap rate, which suggests correlation and mutual information do not necessary select the same features, which is well known from a theoretical perspective as discussed in [27]. Our results corroborate this theory and also suggest that it is reasonable to run both strategies and balance the results. The model comparing relevance across features, *III*, as *Across*, in the tables, shares the lowest overlap rate with the other methods. This method guarantees the least

Overlap	*I*	*II*	*III*	*IV*	*V*	*VI*
I	1	**0.75**	0.25	0.58	0.42	0.42
II	**0.75**	1	0.25	0.58	0.50	0.58
III	0.38	0.38	1	0.50	**0.63**	0.38
IV	**0.70**	**0.70**	0.40	1	0.60	0.50
V	0.50	**0.60**	0.50	**0.60**	1	0.50
VI	0.56	**0.78**	0.33	0.56	0.56	1

Table 7: Overlap rate of 6 algorithms for attack time features

Overlap	*I*	*II*	*III*	*IV*	*V*	*VI*
I	1	0.25	**0.58**	0.33	**0.58**	0.33
II	0.25	1	0.25	0.33	0.33	**0.50**
III	**0.88**	0.38	1	0.63	0.50	0.38
IV	0.40	0.40	**0.50**	1	0.30	0.40
V	**0.70**	0.40	0.40	0.30	1	0.20
VI	0.40	**0.60**	0.30	0.40	0.20	1

Table 8: Overlap rate of 6 algorithms for release time features

mutual information across the selected features, but it does not consider any relation between the features and the label. This is the major difference between this and all the other selection methods. Conversely, the filter model using mutual information, *II*, as Mu_info, in the tables, shares the most overlap with other methods, which shows it is an efficient method on its own.

6 Conclusion and Future Work

In this paper, we introduced a thorough feature set for predicting four of the important DRC parameters. We run a feature selection experiment using six different selection strategies. The final selection is detailed given in Section 5. The results fit the assumption that using a small set of features we are able to reduce noise, computational time and improve the performance at the same time. The results also show that frequency domain features are less efficient when predicting attack and release time, while the opposite is true when predicting threshold and ratio. The results indicate that commonly used features are not sufficient when it comes to predicting the time constant parameters of the DRC. For all four parameters, MFCC related features are the most often selected, which is clearly due to their relations with both frequency information and timbre.

This research aims at devising an intelligent control method for the dynamic range compressor which uses a reference audio [28]. Feature selection results can be used as a guideline for the implementation as well as future research. We will apply a specific feature set when predicting each parameter and focus on the improvement of the design of the selected features. Future work includes conducting a listening test to verify the prediction efficiency and comparing subjective results with numerical test results, as well as optimising the other components of the control system.

References

[1] Verfaille, V., Zölzer, U., and Arfib, D., "Adaptive Digital Audio Effects (A-DAFx): A New Class of Sound Transformations," *IEEE transactions on audio, speech, and language processing*, 14(5), pp. 1817–1831, 2006.

[2] Cartwright, M. B. and Pardo, B., "Social-EQ: Crowdsourcing an Equalization Descriptor Map," in *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.

[3] Stables, R., De Man, B., Enderby, S., Reiss, J., Fazekas, G., and Wilmering, T., "Semantic description of timbral transformations in music production," *Proc. ACM Multimedia, Oct. 15-19, Amsterdam, Netherlands*, pp. 337–341, 2016, doi:10.1145/2964284.2967238.

[4] Ma, Z., De Man, B., Pestana, P. D., Black, D. A., and Reiss, J. D., "Intelligent multitrack dynamic range compression," *Journal of the Audio Engineering Society*, 63(6), pp. 412–426, 2015.

[5] Mason, A., Jillings, N., Ma, Z., Reiss, J. D., and Melchior, F., "Adaptive audio reproduction using personalized compression," in *Audio Engineering Society Conference: 57th International Conference: The Future of Audio Entertainment Technology–Cinema, Television and the Internet*, Audio Engineering Society, 2015.

[6] Mimilakis, S. I., Drossos, K., Virtanen, T., and Schuller, G., "Deep neural networks for dynamic range compression in mastering applications," in *Audio Engineering Society Convention 140*, Audio Engineering Society, 2016.

[7] Kolhoff, P., Preub, J., and Loviscach, J., "Music icons: procedural glyphs for audio files," in *2006 19th Brazilian Symposium on Computer Graphics and Image Processing*, pp. 289–296, IEEE, 2006.

[8] De Man, B., Reiss, J. D., and Stables, R., "Ten Years of Automatic Mixing," in *Proceedings of the 3rd Workshop on Intelligent Music Production*, 2017.

[9] Giannoulis, D., Massberg, M., and Reiss, J. D., "Digital dynamic range compressor design—A tutorial and analysis," *Journal of the Audio Engineering Society*, 60(6), pp. 399–408, 2012.

[10] Sheng, D. and Fazekas, G., "Automatic Control of the Dynamic Range Compressor Using a Regression Model and a Reference Sound," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

[11] Baeza-Yates, R., Ribeiro-Neto, B., et al., *Modern information retrieval*, volume 463, ACM press New York, 1999.

[12] Liu, H. and Yu, L., "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on knowledge and data engineering*, 17(4), pp. 491–502, 2005.

[13] Doraisamy, S., Golzari, S., Mohd, N., Sulaiman, M. N., and Udzir, N. I., "A Study on Feature Selection and Classification Techniques for Automatic Genre Classification of Traditional Malay Music." in *ISMIR*, pp. 331–336, 2008.

[14] Baume, C., Fazekas, G., Barthet, M., Marston, D., and Sandler, M., "Selection of audio features for music emotion recognition using production music," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, Audio Engineering Society, 2014.

[15] Genuer, R., Poggi, J.-M., and Tuleau-Malot, C., "Variable selection using random forests," *Pattern Recognition Letters*, 31(14), pp. 2225–2236, 2010.

[16] Ronan, D., Moffat, D., Gunes, H., Reiss, J. D., et al., "Automatic subgrouping of multitrack audio," in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, 2015.

[17] Martínez Ramírez, M. A. and Reiss, J. D., "Stem audio mixing as a content-based transformation of audio features," in *19th International Workshop on Multimedia Signal Processing (MMSP), IEEE*, 2017.

[18] Stone, C. J., Friedman, J., Breiman, L., and Olshen, R., "Classification and regression trees," *Wadsworth International Group*, 8, pp. 452–456, 1984.

[19] Zölzer, U., Amatriain, X., and Arfib, D., *DAFX: digital audio effects*, volume 1, Wiley Online Library, 2002.

[20] Peeters, G., "A large set of audio features for sound description (similarity and classification) in the CUIDADO project," 2004.

[21] Bennasar, M., Hicks, Y., and Setchi, R., "Feature selection using joint mutual information maximisation," *Expert Systems with Applications*, 42(22), pp. 8520–8532, 2015.

[22] Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R., "RWC Music Database: Music Genre Database and Musical Instrument Sound Database," *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, pp. 229–230, 2003.

[23] Stables, R., Enderby, S., Man, B., Fazekas, G., Reiss, J. D., et al., "SAFE: A system for the extraction and retrieval of semantic audio descriptors," *15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, ????

[24] Benesty, J., Chen, J., Huang, Y., and Cohen, I., "Pearson correlation coefficient," in *Noise reduction in speech processing*, pp. 1–4, Springer, 2009.

[25] Vinh, N. X., Epps, J., and Bailey, J., "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *Journal of Machine Learning Research*, 11(Oct), pp. 2837–2854, 2010.

[26] Peng, H., Long, F., and Ding, C., "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, 27(8), pp. 1226–1238, 2005.

[27] Li, W., "Mutual information functions versus correlation functions," *Journal of statistical physics*, 60(5-6), pp. 823–837, 1990.

[28] Sheng, D. and Fazekas, G., "Feature design using audio decomposition for intelligent control of the dynamic range compressor," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP18)*, 2018.