A Practical Singing Voice Detection System Based on GRU-RNN

Zhigao Chen, Xulong Zhang, Jin Deng, Juanjuan Li, Yiliang Jiang and Wei Li

Abstract In this paper, we present a practical three-step approach for singing voice detection based on a gated recurrent unit (GRU) recurrent neural network (RNN) and the proposed method achieves comparable results to state-of-the-art method. We combine four classic features—namely Mel-frequency Cepstral Coefficients (MFCC), Mel-filter Bank, Linear Predictive Cepstral Coefficients (LPCC), and Chroma. Then, the mixed signal is first preprocessed by singing voice separation (SVS) with the Deep U-Net Convolutional Networks. Long short-term memory (LSTM) and GRU are both proposed to solve the gradient vanish problem in RNN. In our experiments, we set the block duration as 120 ms and 720 ms respectively, and we get comparable or better results than results from state-of-the-art methods, while results on Jamendo are not as good as those from RWC-Pop.

Keywords Singing voice detection (SVD) · Gated recurrent unit (GRU) · Recurrent neural network (RNN) · Music information retrieval (MIR)

Z. Chen · X. Zhang · J. Deng · J. Li · Y. Jiang · W. Li (✉)
Department of Computer Science, Fudan University, 201203 Shanghai, China
e-mail: weili-fudan@fudan.edu.cn

Z. Chen
e-mail: zgchen15@fudan.edu.cn

X. Zhang
e-mail: xlzhang14@fudan.edu.cn

J. Deng
e-mail: jdeng17@fudan.edu.cn

J. Li
e-mail: jjli16@fudan.edu.cn

Y. Jiang
e-mail: yljiang17@fudan.edu.cn

W. Li
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, 201203
Shanghai, China

© Springer Nature Singapore Pte Ltd. 2019

W. Li et al. (eds.), *Proceedings of the 6th Conference on Sound and Music Technology (CSMT)*, Lecture Notes in Electrical Engineering 568,
https://doi.org/10.1007/978-981-13-8707-4_2

1 Introduction

Singing voice detection (SVD) aims to localize portions of sound that containing human voice. Currently in music information retrieval (MIR), SVD is receiving increasing concerns due to its great usefulness in some singer related tasks—e.g., singer identification [1, 2], melody extraction [3].

Singing voice separation (SVS) was not frequently used as a pretreatment in SVD in many studies. However, Hennequin [4] used double stage Harmonic/Percussive Sound Separation (HPSS) [5], a simple method to separate monaural audio into harmonic and percussive components and to extract features, and got remarkable results compared with others. We choose the Deep U-Net Convolutional Networks proposed by Jansson [6], to separate singing voice directly and use the vocal signal to complete the next steps.

Traditional statistical methods with widely used speech features have been applied to SVD [7]. Following the traditional framework, they extracted a set of features and fed them to a classifier—e.g., support vector machines (SVMs) [8, 9] and random forests [10, 11]. A singing voice typically involves a higher pitch than regular speech, with wide, exaggerated intonation changes. These studies [12, 13] were often based on commonly used features in speech processing field—e.g., Mel-frequency Cepstral Coefficients (MFCC) and Linear Predictive Cepstral Coefficients (LPCC), which may not be sufficient to capture the characteristics of the accompanied singing.

It's a common phenomenon that the same features will achieve different performance in different datasets. Instead of using well-designed complex handcraft features [14], we combine four classic features—MFCC, Mel-filter Bank [4], LPCC, and Chroma [15] from MIR—and tried to depict audio characteristics more comprehensively with those features above.

GRU [16] is able to take temporal context into consideration, hence it should achieve a better performance than traditional machine learning techniques. Additionally, GRU solves the problem of gradient vanish with a simpler structure than LSTM—thus, it is preferable for real-time application. Results show that, without using temporal smoothing for post-process, our GRU-RNN can perform as well as state-of-the-art methods on public datasets.

In sum, we present a practical three-step approach for SVD in this paper. First, we separate singing voice with Deep U-Net Convolutional Networks. Second, we extract MFCC, Mel-filter Bank, LPCC, and Chroma as features. Third, we use GRU-RNN as the classifier.

This paper is organized as follows. Section 2 presents related work of this task, Sect. 3 outlines our method, Sect. 4 describes our results, and Sect. 5 presents our conclusions.

2 Related Work

SVD aims at marking out audio segments that contain human voices, which includes singing and speech, as a matter of fact. We will introduce some of the principal methods from previous studies in this section, and we'll compare our results to theirs in Sect. 4.

In an early study, Rocamora and Herrera [17] compared existing descriptors with a statistical classifier. It came out that MFCC achieved the best performance in their experiment by using their private dataset—the accuracy was 78.5%.

Ramona [9] performed this task with a large feature set, a support vector machine (SVM), and a temporal smoothing method with the Hidden Markov Model (HMM). They achieved 82% accuracy in their experiment. Mauch et al. [18] used four timbre and melody features in different combinations and fed them to support vector machine with the Hidden Markov Model (SVM-HMM) to perform the task. Their results demonstrated that the top accuracy is 87.2% when all four features were used.

Lehner et al. [10] proposed a towards light-weight and real-time online SVD system. They used only simple optimized MFCCs as the feature and the optimized random forest as the classifier. They achieved 82.36% accuracy, after manually adjusting the features and classifiers.

Observing that one of the biggest problems in automatic SVD is the confusion between vocals and instruments, based on the work of [10], Lehner et al. [11] designed a set of new audio features to reduce the amount of false vocal detections. The features consisted of fluctograms, vocal variances, spectral flatness and spectral contractions. With the new hard-craft features, results appeared to be, at least, on par with more complex state-of-the-art methods with common features.

Lehner introduced the LSTM-RNN into SVD in [14]. Different from their work in [10], which only used simple MFCCs, they combine 30 MFCCs, their delta coefficients, and other three spectral features—totaling 111 attributes. They achieved state-of-the-art results on the two publicly available databases—namely Jamendo and RWC-Pop. Furthermore, Eyben et al. [19] proposed a data-driven approach based on LSTM-RNN and standard RASTA-PLP frontend features, and results showed that LSTM-RNN outperformed all other statistical baselines. Leglaive [4] used a Bidirectional LSTM-RNN (BLSTM-RNN) as the classifier. They used just Mel-filter Bank preprocessed by HPSS, and they achieved an accuracy of 91.5% on public datasets Jamendo.

In Schlüter's work [20], pitch shifting, time stretching, and random frequency filtering were used to augment the training datasets on the public datasets Jamendo [9] and RWC-Pop [18], and the CNN model was used on Mel spectrograms to design the SVD system. Finally, the prosed method by Schlüter achieved an error rate of around 9%, which is on par with state-of-the-art results.

3 Method

We propose a three-step system. The audio signal is first preprocessed by SVS, then we extract features and fed them to the classifier. The dataset is divided into two parts, training set and testing set, and they are independent of each other. The network is first trained by the training set, then used to predict the label of the testing set. The overview of our SVD system is shown in Fig. 1. Details of every step are discussed below.

3.1 Singing Voice Separation

The audio signal is first preprocessed by SVS—we can split the mixed music signal into vocal signal and accompaniment signal through SVS. We use the Deep U-Net Convolutional Networks in [6] to accomplish this task. Datasets that we used to train the U-Nets were iKala [21] and MedleyDB [22]. The main steps are as follows.

1. Train two U-Nets respectively to predict vocal and instrumental spectrogram masks. The U-Net operates exclusively on the magnitude of audio spectrograms.
2. Compute the spectrogram mask of signal with well-trained U-Net. Apply the mask to the magnitude of original spectrum.
3. Reconstruct the signal with the new magnitude and original phase.

3.2 Feature Extraction

We describe all four features that we used in our experiment here. Many features were applied to SVD task: we picked four classic features from both speech and music field, and they were MFCC [14], Mel-filter Bank, LPCC [23], and Chroma [15].

MFCC is widely used in many speech- and audio-related tasks [17], as it characterizes the timbre of the human voice. It was found that MFCC achieved the best performance in [17]. Mel-filter Bank was extracted from a filter bank on Mel scale,

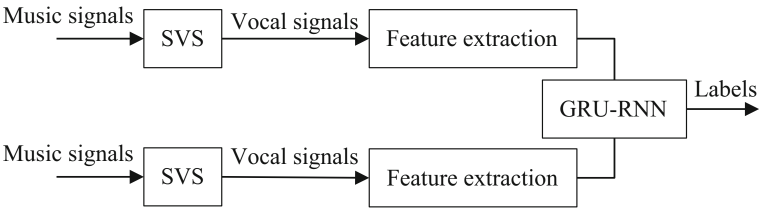

Fig. 1 A system overview

and it had a good performance in SVD [4]. LPCC had a good relationship with the vocal tract of the speaker, and it was calculated by introducing the cepstrum coefficients to the LPC parameters. Chroma, also known as Harmonic Pitch Class Profile, collects spectral energy supporting each semitone of the octave and could consider the timbre of music. It's a well-established tool for analyzing and comparing music signals [24]. We attempted to depict audio characteristics more comprehensively with the aforementioned features.

In our experiment, the frame length was set to 40 ms, with overlapping of 20 ms. On the short-scale frames stated above, we extracted 20-order MFCC, 20-order Mel-filter Bank, 12-order LPCC, and 12-order Chroma separately, then combined them to get a 64-dimensional feature.

3.3 GRU-RNN

As we know, temporal context is sometimes needful for human beings to make a vocal-nonvocal decision, and RNN could take the temporal context into consideration in classification tasks [25], so that it is appropriate to use here. The major problem of the traditional RNN is that gradient vanish, GRU, and LSTM do not solve this problem in the same way [16]. They both add the update from t to t + 1 inside these units—that's the most prominent feature that they share. In other words, both GRU and LSTM will add new content to existing content.

GRU and LSTM have a number of differences. GRU exposes its full content without any control, while LSTM controls the exposure of the memory content—i.e., GRU is simpler in structure than LSTM. Another difference is the updating of new memory content. The control of information flow in GRU is tied via the update gate, while LSTM is via the forget gate independently.

Experiments in [16] showed that convergence of GRU is often faster and the final solutions tend to be better than LSTM. Encouraged by this, we present a unidirectional RNN with a hidden layer, which consists of 60 GRU units. The input is shaped as the dimension of the combined feature and multiplied by a fixed duration block [10], and the block duration will be tuned in Sect. 4. An output layer with a single sigmoid is added. The output of the classifier is 1 or 0, 1 indicating singing and 0 indicating non–singing. Dropout is set as 0.2, and early stopping strategy is used here—it will stop if the loss of validation data gets no improvement over 5 epochs.

4 Result

In this section, we present the results on two available public datasets—i.e., RWC-Pop and Jamendo. We train several GRU-RNNs according to the previous strategy and compare the results to some methods in Sect. 2. Then, we calculate four frequently-used evaluation index [27]—i.e., frame-wise accuracy, precision, recall, and F1-measure.

4.1 Datasets

RWC Popular Music Dataset. The RWC-Pop dataset consists of 100 pop songs, with annotations that Mauch et al. [18] released. It contains 80 Japanese pop songs and 20 English pop songs. The audio files are stereo, with a sampling rate of 44.1 kHz, and the sampling precision of 16 bits. We converted the stereo files to mono wav first. The whole set is well balanced, since 51.2% of frames are singing segments and 48.8% are non-singing segments. For better comparison, we conducted the 5-fold cross validation [11]. All the data are divided into five parts: one for testing and the other four for training. The validation set is extracted 20% data from the training set. The three are independent of each other.

Jamendo Corpus. The Jamendo Corpus consists of 93 songs from Jamendo free music sharing platform with Creative Commons License. They were annotated by the same person [9]. The audio files are stereo, with a sampling rate of 44.1 kHz, Vorbis OGG format, with the sampling precision of 112 KB/s, or MP3 format, with the sampling precision of 128 KB/s. We converted the stereo files to mono wav first. The whole set is well balanced, since 50.3% of frames are singing segments and 49.7% are non-singing segments. All the data are divided into three independent parts, the same as in [9]—i.e., training set, validation set, and testing set, respectively containing 61, 16, and 16 songs.

4.2 Evaluations

For a better comprehensive evaluation, we compare predicted results of testing set to the ground truth labels and get the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Then, we calculate four frequently-used evaluation index [26]—i.e., frame-wise accuracy, precision, recall, and F1-measure.

4.3 Results and Discussion

Comparison between mixed signals and separated vocal signals. We compare the results of mixed signals and separated vocal signals in this experiment. Mixed signals are original mono signals. As previously mentioned, Deep U-Net Convolutional Networks were used to separate the vocal from the raw data. We use the combination of all the features mentioned and the parameters of GRU-RNN in Sect. 3. Block duration is set as 25 frames.

As we can see in Fig. 2, the performance in both datasets are significantly improved after the SVS. It raised the accuracy and F1-measure, by around 10% in RWC-Pop dataset and by around 15% in Jamendo dataset. So, it is useful to preprocess with

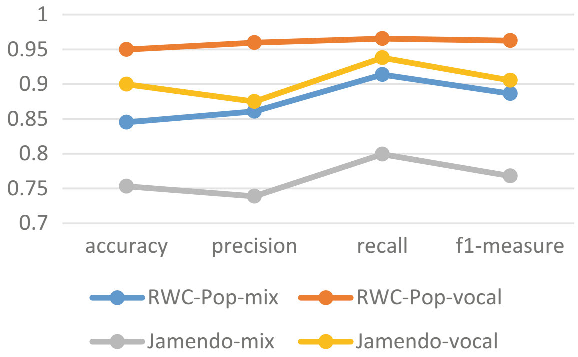

Fig. 2 Comparison between mixed signals and separated

SVS in our task here. As a matter of fact, eliminating or reducing the impacts of the accompaniment could be very helpful in various related tasks.

Fine-tuning of block duration. In previous studies, different block durations were used for the decision. Also, in our experiments, on the premise of the above parameters mentioned in Sect. 3, we found that the block duration has a great influence on our SVD system. So, we compare the results of different durations and choose an appropriate value for our experiments.

The results show that there is a positive correlation between the performance and the block duration. However, while this may seem obvious, annotation precision decreases with increased block duration. There are no black-and-white lines for annotation precision, and it is not necessarily the same in different tasks, but it is often determined by experiences.

Various durations were chosen in different studies: we choose two representative state-of-the-art methods for further comparison. Lehner [14] achieved better performance on RWC-Pop dataset, while Hennequin [4] did better with the Jamendo dataset—in fact, Hennequin [4] compared their performance with previous studies on Jamendo only. The block duration of Lehner [14] was 140 ms, and Hennequin [4] sets its duration as 800 ms. So, we finally decide to choose 120 and 720 ms to compare our results with others.

Final results. Table 1 shows our experimental results on RWC-Pop, compared with Mauch [18], Schlüter [20], Lehner-1 [11], Lehner-2 [10], Lehner-3 [14]. We set the block duration as 120 and 720 ms—they were called GRU-RNN-1 and GRU-RNN-2 respectively. Combining the results of previous studies, the best accuracy, precision, recall and F1-measure are 0.927, 0.938, 0.935, and 0.936 respectively. GRU-RNN-1 gets the best performance in recall and F1-measure and is slightly less than state-of-the-art methods on RWC-Pop. GRU-RNN-2 outperforms state-of-the-art methods on this dataset, about 3–4% above it. Given the fact that we use the combination of low-level features directly without any post-processing, our results are remarkable. Moreover, our GRU-RNN has a simpler structure than LSTM—thus, it has higher computational efficiency – so, our method is better for real-time applications than Lehner-3 [14].

Table 1 Experiment of block duration on RWC-Pop

Frames	Duration (ms)	Accuracy	Precision	Recall	F1-measure
5	**120**	**0.9205**	**0.9267**	**0.9542**	**0.9402**
7	160	0.9136	0.9080	0.9665	0.9363
11	240	0.9265	0.9271	0.9647	0.9455
17	360	0.9412	0.9514	0.9609	0.9562
25	520	0.9477	0.9560	0.9665	0.9612
35	**720**	**0.9531**	**0.9605**	**0.9696**	**0.9650**
47	960	0.9592	0.9654	0.9728	0.9691
61	1240	0.9594	0.9699	0.9671	0.9685
77	1560	0.9663	0.9726	0.9736	0.9731
95	1920	0.9617	0.9681	0.9681	0.9681
115	2320	0.9664	0.9609	0.9800	0.9704
137	2760	0.9682	0.9738	0.9644	0.9691
161	3240	0.9711	0.9765	0.9629	0.9696

Table 2 Results on RWC Popular Music Dataset

	Accuracy	Precision	Recall	F1-measure
Mauch [18]	0.872	0.887	0.921	0.904
Schlüter [20]	**0.927**	–	**0.935**	–
Lehner-1 [11]	0.875	0.875	0.926	0.900
Lehner-2 [10]	0.868	0.879	0.906	0.892
Lehner-3 [14]	0.923	**0.938**	0.934	**0.936**
GRU-RNN-1	*0.9205*	*0.9267*	***0.9542***	***0.9402***
GRU-RNN-2	***0.9531***	***0.9605***	***0.9696***	***0.9650***

Table 2 shows our experimental results on Jamendo, compared with Ramona [9], Schlüter [20], Lehner-1 [11], Lehner-2 [10], Lehner-3 [14], Leglaive [25]. We set the block duration as 120 and 720 ms, they were called GRU-RNN-1 and GRU-RNN-2 respectively.

Results on Jamendo are not as good as on RWC-Pop, as Table 3 shows. Combining the results of previous studies, the best accuracy, precision, recall and F1-measure are 0.923, 0.898, 0.926, and 0.910 respectively. GRU-RNN-1 gets the best performance in recall, which is not as good as the best results of previous studies—namely so-called state-of-the-art results on Jamendo. GRU-RNN-2 does the best in recall and F1-measure, while it is about 1% less than the best in accuracy and precision. But,

Table 3 Results on Jamendo Corpus

	Accuracy	Precision	Recall	F1-measure
Ramona [9]	0.822	–	–	0.831
Schlüter [20]	**0.923**	–	0.903	–
Lehner-1 [11]	0.882	0.880	0.862	0.871
Lehner-2 [10]	0.848	–	–	0.846
Lehner-3 [14]	0.894	**0.898**	0.906	0.902
Leglaive [4]	0.915	0.895	**0.926**	**0.910**
GRU-RNN-1	*0.8821*	*0.8539*	***0.9278***	*0.8893*
GRU-RNN-2	*0.9082*	*0.8923*	***0.9331***	***0.9122***

in general, our method gets comparable performance compared with state-of-the-art methods. There is one thing we must notice here, Schlüter [20] used pitch shifting and time stretching to make the data augmentation, while other studies were performed on original dataset. Data volume is a significant factor for machine learning, so it does not seem fair to compare our paper with Schlüter's work [20], but we still put it here for reference.

5 Conclusion

In this paper, we propose a practical three-step approach, which means good performances, simple feature selection, and higher computational efficiency for SVD based on a gated recurrent unit (GRU) recurrent neural network (RNN).

These steps are SVD, feature extraction and pattern recognition. GRU is able to take temporal context into consideration and our features are easy to extract and combine. We abandon the regular post-processing step – namely temporal smoothing.

We set the block duration as 120 and 720 ms respectively and get comparable or better performances to state-of-the-art methods with different parameters. As one can see, results on Jamendo are not as good as on RWC-Pop. Most of all, our GRU has a simpler structure and higher computational efficiency than LSTM, so it is better for real-time applications.

Future work includes the following aspects. Use the bidirectional-GRU instead of our unidirectional GRU. Find the reason why results on Jamendo are always worse than RWC-Pop and make specific improvements.

Acknowledgements This research was supported by NSFC 61671156. We thank our colleagues from Fudan University, who provided insight and expertise that greatly assisted the research, although they may not agree with all the interpretations of this paper.

References

1. Leglaive S, Hennequin R, Badeau R (2015) Singing voice detection with deep recurrent neural networks. In: Proceeding of IEEE international conference on acoustics, speech and signal processing (ICASSP). Brisbane, Australia, pp 121–125
2. Kim YE, Whitman B (2002) Singer identification in popular music recordings using voice coding features. In: Proceedings of the 3rd international conference on music information retrieval. Paris, France, pp 13–17
3. Salamon J, Gómez E (2012) Melody extraction from polyphonic music signals using pitch contour characteristics. IEEE Trans Audio Speech Lang Process 20(6):1759–1770
4. Leglaive S, Hennequin R, Badeau R (2015) Singing voice detection with deep recurrent neural networks. In: Proceeding of IEEE international conference on acoustics, speech and signal processing (ICASSP). Brisbane, Australia, pp 121–125
5. Ono N, Miyamoto K, Le Roux J, Kameoka H, Sagayama S (2008) Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. In: Proceeding of 16th European signal processing conference. Lausanne, Switzerland
6. Jansson A, Humphrey E, Montecchio N, Bittner R, Kumar A, Weyde T (2017) Singing voice separation with deep U-Net convolutional networks. In: Proceeding of 18th international society for music information retrieval conference. Suzhou, China
7. Sonnleitner R, Niedermayer B, Widmer G, Schlüter J (2012) A simple and effective spectral feature for speech detection in mixed audio signals. In: Proceedings of the 15th international conference on digital audio effects (DAFx'12). York, UK
8. Vembu S, Baumann S (2005) Separation of vocals from polyphonic audio recordings. In: Proceeding of international society for music information retrieval conference, London, UK, pp 337–344
9. Ramona M, Richard G, David B (2008) Vocal detection in music with support vector machines. In: Proceeding of IEEE international conference on acoustics, speech and signal processing (ICASSP). Las Vegas, USA, pp 1885–1888
10. Lehner B, Sonnleitner R, Widmer G (2013) Towards light-weight, real-time-capable singing voice detection. In: Proceeding of international society for music information retrieval conference. Curitiba, Brazil, pp 53–58
11. Lehner B, Widmer G, Sonnleitner, R (2014) On the reduction of false positives in singing voice detection. In: Proceeding of IEEE international conference on acoustics, speech and signal processing. Florence, Italy, pp 7480–7484
12. Regnier L, Peeters G (2009) Singing voice detection in music tracks using direct voice vibrato detection. In: Proceeding of IEEE international conference on acoustics, speech and signal processing. Taipei, Taiwan, pp 1685–1688
13. Pikrakis A, Kopsinis Y, Kroher N, Díaz-Báñez JM (2016) Unsupervised singing voice detection using dictionary learning. In: Proceeding of 24th European signal processing conference. Budapest, Hungary, pp 1212–1216
14. Lehner B, Widmer G, Bock S (2015) A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks. In Proceeding of 23rd European signal processing conference. Nice, France, pp 21–25
15. Ellis DPW, Poliner GE (2007) Identifying cover songs' with chroma features and dynamic programming beat tracking. In: Proceeding of IEEE international conference on acoustics, speech and signal processing. Honolulu, USA, pp 1429–1432

16. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint, 1412.3555
17. Rocamora M, Herrera P (2017) Comparing audio descriptors for singing voice detection in music audio files. In: 11th Brazilian symposium on computer music. São Paulo, Brazil, pp 27–36
18. Mauch M, Fujihara H, Yoshii K, Goto M (2011) Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In: Proceeding of international society for music information retrieval conference. Miami, Florida, pp 233–238
19. Eyben F, Weninger F, Squartini S, Schuller B (2013) Real-life voice activity detection with LSTM recurrent neural networks and an application to hollywood movies. In: Proceeding of IEEE international conference on acoustics, speech and signal processing. Vancouver, Canada, pp 483–487
20. Schlüter J, Grill T (2015) Exploring data augmentation for improved singing voice detection with neural networks. In: Proceeding of international society for music information retrieval conference. Malaga, Spain, pp 121–126
21. Chan TS, Yeh TC, Fan ZC, Chen HW, Su L, Yang YH, Jang R (2015) Vocal activity informed singing voice separation with the iKala dataset. In: Proceeding of 2015 IEEE international conference on acoustics, speech and signal processing. Brisbane, Australia, pp 718–722
22. Bittner RM, Salamon J, Tierney M, Mauch M, Cannam C, Bello JP (2014) MedleyDB: a multitrack dataset for annotation-intensive MIR research. In: Proceeding of international society for music information retrieval conference, vol 14. Taipei, Taiwan, pp 155–160
23. Gupta H, Gupta D (2016) LPC and LPCC method of feature extraction in speech recognition system. In: Proceeding of 6th international conference cloud system and big data engineering. Noida, India, pp 498–502
24. Muller M, Ewert S, Kreuzer S (2009) Making chroma features more robust to timbre changes. In: Proceeding of IEEE international conference on acoustics, speech and signal processing. Taipei, Taiwan, pp 1877–1880
25. Leglaive S, Hennequin R, Badeau R (2015) Singing voice detection with deep recurrent neural networks. In: Proceeding of IEEE international conference on acoustics, speech and signal processing (ICASSP). Brisbane, Australia, pp 121–125
26. Sokolova M, Lapalme G (2009) A systematic analysis of performance measures for classification tasks. Inf Process Manage 45(4):427–437