2021-03-31

GLOTTAL CLOSURE INSTANT DETECTION USING ECHO STATE NETWORKS

Steiner, P

Studientexte zur Sprachkommunikation Band 99: Elektronische Sprachsignalverarbeitung 2021
Conference proceedings of the 32st conference in Berlin with 41 contributions. ISBN:
978-3-959082-27-3

GLOTTAL CLOSURE INSTANT DETECTION USING ECHO STATE NETWORKS

Peter Steiner[1], *Ian S. Howard*[2], *Peter Birkholz*[1]

[1]*Institute of Acoustics and Speech Communication, Technische Universität Dresden*
[2]*Centre for Robotics and Neural Systems, University of Plymouth, Plymouth, PL4 8AA, UK*
peter.steiner@tu-dresden.de

Abstract: The Time of Excitation (Tx) of speech, also widely known as the Glottal Closure Instants (GCI) denote the points in time at which the vocal folds close during the production of voiced speech. In this paper, we extend a previous approach based on a multilayer perceptron (MLP) using Echo State Networks (ESN), a variant of a Recurrent Neural Network (RNN). We show that the MLP and ESN approaches lead to similar results. The ESN model performed better than the MLP when the latter used only a single input sample (0.86 vs 0.75 area under the ROC plot), whereas the MLP slightly outperformed the ESN (0.98 vs 0.97 area under the ROC plot) when its was provided with a sufficient number of surrounding speech samples.

1 Introduction

During the production of voiced sounds, the vocal folds are adducted in a way that airflow leads them to oscillate. This leads to modulation of the air stream from the lung into a quasiperiodic flow of air-pulses. This excitation signal is acoustically filtered in the remaining vocal tract and can lead to the production of voiced phones, such as vowels or nasals. The points in time at which the vocal folds close during the production of voiced speech are known as the Glottal Closure Instants (GCI).

In speech processing, GCI detection is useful for many tasks, including vocal disorder analysis [1, 2] and fundamental frequency (f_0) estimation [3, 4]. The fundamental period T_0 can be defined as the time interval between successive vocal folds opening and closures and thus corresponds to the time between two adjacent GCIs. Initial work made use of MLP analysis of speech, labelled with Electroglottography (EGG), to detect GCIs directly from the acoustic speech signal [3].

More recently, Reddy et al. [2] explored the properties of Electroglottography (EGG) signals for four different speech disorders and showed that each type of disorder comes along with different properties of the EGG signal, such as the open quotient of the larynx. GCI detection is used as starting point for this kind of analysis to provide a robust period-synchronous segmentation of speech signals. GCI estimation is also used in other applications, such as speech synthesis, which also depends on overlapping and adding speech samples period-synchronously [5].

In this paper, we use the dataset from [4] that contains audio data and Electroglottography (EGG) signals as a source of ground truth, and present an approach to estimate GCIs from raw speech signals using Echo State Networks (ESNs). ESN is a variant of a Recurrent Neural Network (RNN), that has recently achieved state-of-the-art results in note onset detection [6, 7], which is a related task to that of GCI detection.

Here we investigate several aspects of such a neural GCI detector, including the effect of temporal context, the impact of non-linearity and recurrent connections on the result, and

causal vs. non-causal operation. We show that a causal ESN model outperforms the baseline multilayer perceptron (MLP) [4] (possibly due to its recurrent connections) and that it is relatively noise-robust and can generalize well to conditions not experienced in the training dataset. We note that non-causal bidirectional architectures (without an online capacity) are able to achieve even better results.

2 Glottal Closure Instance Estimation with Echo State Networks

2.1 Input data preparation

For the experiments presented here, we directly utilized the raw speech signal $s[k]$ with the sample index k and a sampling frequency of 4 kHz. We considered two different settings:

1. Online GCI estimation: The ESN received exactly one speech sample as input.
2. Context-based GCI estimation: The ESN received a window of speech samples centered around the current target value as input.

We did not further normalize or standardize the raw speech samples but rather directly fed them into the ESN either as a single speech sample at a time, or in terms of feature vectors consisting of a window over multiple speech samples.

2.2 Echo State Network (ESN)

The ESN architecture used here is largely adapted from [7]. It consists of the input weights $\mathbf{W}^{\text{in}}$, the reservoir weights $\mathbf{W}^{\text{res}}$ and the output weights $\mathbf{W}^{\text{out}}$.

The input weight matrix $\mathbf{W}^{\text{in}}$ has the dimension of $N^{\text{res}} \times N^{\text{in}}$ where N^{in} is determined by the number of speech samples that the ESN receives in one time step. N^{res} is the size of the reservoir. All values in this matrix were initialized from a uniform distribution between ± 1.0. Each node of the reservoir was then connected to only the K^{in} randomly selected input entries. The other connections were set to zero, leading to a very sparse matrix $\mathbf{W}^{\text{in}}$. As we are working with different N^{in} and it was shown in [8], that K^{in} should be greater than 2, we use Equation (1) to define $K^{\text{in}} = 5$ for larger windows. This is a common choice.

$$K^{\text{in}} = \begin{cases} N^{\text{in}} & \text{if } N^{\text{in}} \leq 5 \\ 5 & \text{else.} \end{cases} \tag{1}$$

The input weight matrix was then scaled using the input scaling factor α_{U}, which constitutes a hyper-parameter that needs to be tuned.

The reservoir weight matrix $\mathbf{W}^{\text{res}}$ specifies the recurrent connections inside the reservoir, in which all nodes are essentially connected to each other. After initializing the reservoir weight matrix by sampling from a standard normal distribution, we limited the number of connections so that each reservoir node received values from only $K^{\text{rec}} = 10$ other nodes that were selected randomly. Next, the reservoir matrix $\mathbf{W}^{\text{res}}$ was normalized by its largest absolute eigenvalue to achieve a spectral radius $\rho = 1.0$, since it was shown in [9] that the echo state property holds as long as $\rho \leq 1.0$. By tuning α_{U} and ρ, it is possible to balance how strongly the network memorizes past inputs compared to the present input.

If $\mathbf{r}[n]$ represents the reservoir state, the basic equations to describe the ESN can be written as follows:

$$\mathbf{r}[n] = (1-\lambda)\mathbf{r}[n-1] + \lambda f_{\text{res}}(\mathbf{W}^{\text{in}}\mathbf{u}[n] + \mathbf{W}^{\text{res}}\mathbf{r}[n-1]) \tag{2}$$

$$y[n] = \mathbf{W}^{\text{out}} \cdot \mathbf{r}[n] \tag{3}$$

Equation (2) describes how the speech signal is fed into the ESN and how the current reservoir state is computed based on the input features and the last reservoir state. The reservoir activation function $f_{\text{res}}(\cdot)$ controls the non-linearity of the system. Here, the tanh-function was used.

Equation (3) shows how to compute the one-dimensional output $y[n]$ from a given reservoir state $\mathbf{r}[n]$, which was expanded by one bias term. The output is obtained by a linear combination of the reservoir state and the output weights $\mathbf{W}^{\text{out}}$. For training, all reservoir states were collected in the reservoir state collection matrix $\mathbf{R}$, and augmented by a single bias term. The target outputs $d[n]$, which were 0.0 for non-GCIs, 0.5 before and after GCIs, and 1.0 for GCIs were collected into the desired output collection vector $\mathbf{d}$. Afterwards, $\mathbf{W}^{\text{out}}$ was obtained using regularized linear regression (4), i.e. ridge regression to prevent overfitting to the training data. The regularization parameter $\varepsilon = 0.001$ penalized large values in $\mathbf{W}^{\text{out}}$, and $\mathbf{I}$ is the identity matrix.

$$\mathbf{W}^{\text{out}} = \left(\mathbf{R}\mathbf{R}^{\text{T}} + \varepsilon\mathbf{I}\right)^{-1}\left(\mathbf{D}\mathbf{R}^{\text{T}}\right) \tag{4}$$

The size of the output weight matrix $N^{\text{out}} \times (N^{\text{res}} + 1)$ determines the total number of free parameters to be trained in ESNs. The output $\mathbf{y}[n]$ corresponded to the GCI estimation function.

Bidirectional reservoirs

In the case of bidirectional reservoirs, the input was first fed through the ESN as described before. Before the linear regression, the inputs were reversed in time, and again fed into the same reservoir. Afterwards, the reservoir states were again reversed in time. The reservoir state collection matrix $\mathbf{R}$ was finally built by combining the states from the forward and backward passes. This doubled the number of free parameters for the linear regression. For example, in the unidirectional ESN the number of features for a reservoir with 500 neurons is 500 in comparison with 1000 in the bidirectional case. The final output training was carried out as before.

3 Experimental setup

3.1 Dataset

In this paper, we used the dataset from [4] that contains audio data and Electroglottography (EGG) signals as ground truth. Both the raw speech and EGG signals have a sampling frequency of 32 kHz. We down-sampled both speech and EGG signals to 4 kHz.
The ground truth was generated from the EGG signals as follows:

1. Compute the first derivative of the EGG signals over time.
2. Set all values in the first derivative below 0.04 to zero.
3. Find all peaks in the rectified EGG signal.

4. Binarize the rectified EGG signal – peaks are 1, remaining samples 0.

This preprocessing is directly taken from [4] and leads to relatively accurate annotations, although better schemes, including ones involving interaction and hand-checking by human operators are possible and discussed in [3]. We note that the ground truth was not entirely reliable and there were several spurious false-alarm annotations where it was unlikely that GCIs would occur, especially at the beginning and end of voiced utterances.

The dataset contains in total 56 male students, who all read the phonetically balanced "Arthur the Rat" passage in British English. We follow [4] and used eight speakers for training and optimizing the ESN models (954 s audio signal duration). We did not use more speakers to limit the amount of training time. We used four more speakers as an unseen test set (426 s audio signal duration) that was not used for any training or optimization steps. To investigate the effect of noise, we adapted the ESN models with a noisy version of the training set by overlapping its utterances with Gaussian noise and a Signal-to-Noise-Ratio (SNR) of 0 dB. The evaluation was then carried out with the SNR-values of 100 dB, 20 dB, 0 dB, -3 dB, -10 dB and -20 dB.

3.2 Measurements

We utilize the Receiver Operating Characteristic (ROC) graph [10], which is a simple technique to compare different classifiers. It was already used in [3, 4] to indicate the performance of GCI algorithms. For a binary classifier, the true positive rate (TPR) and false positive rate (FPR) are evaluated for different thresholds and subsequently visualized. If a detector works very well, then the area under the ROC plot is close to 1.0, and the classifier is very robust. If the area under the ROC plot is close to 0.5, it means that TPR and FPR are almost equal for all threshold values and the classifier only exhibits random performance. If TPR is less than FPR, this indicates that the classifier is doing something fundamentally inappropriate.

3.3 Implementation and optimization strategy

All algorithms used for the experiments were implemented in Python 3. The ESN models were trained using the newly developed PyRCN library[1]. For audio signal processing, we used the librosa library [11].

To optimize the hyper-parameters of our different ESN models, we use the same approach as proposed in [7, 12]:

1. Jointly optimize input scaling α_{U} and spectral radius ρ: This optimization balances a trade-off between forward and recurrent connections.

2. Optimize leakage λ: This optimization is responsible to determine the amount of leaky integration. If λ is very small, the ESN output is rather smooth. Otherwise, it can change fast.

The optimized values were $\alpha_{\mathrm{U}} = 14.6$, $\rho = 1.0$ and $\lambda = 1.0$, and can be interpreted as we should have normalized the audio signal, as the input scaling factor is very high. The large spectral radius makes sense as the ESN needs a lot of memory in order to compute the relationship between the current and past inputs. The fact that no leaky integration was required for this task can be understood as the output needs to change very fast. In the ideal case, the output would be zero for all the time but for GCIs. In reality, the values are neither binary nor bounded between zero and one.

[1] https://github.com/TUD-STKS/PyRCN

4 Results

In this section, we progressively evaluate different models for GCI estimation. We start with a comparison of a ridge regression, MLP and an ESN model to show the improvement by adding free parameters and a non-linear activation function as well as recurrent connections.

4.1 Comparison of different model architectures

Fig. 1 shows the receiver operating characteristic (ROC) analysis of different model architectures and temporal contexts. All models, except the bidirectional ESN (see Figure 1d), strongly benefit from increased window sizes. This means that the task of GCI estimation is strongly context-dependent. Figure 1a shows that the ridge regression model performed worse than all other models. This result is not that surprising, since this model is linear and has very few trainable parameters. However, we can see that the results for larger window sizes strongly improve. Comparing that to Figure 1b, we can observe that the model architecture from [4] strongly outperformed the linear model, especially for small window sizes. Also for the linear model, we see a strong improvement with an increased window size. Although we have used exactly the same MLP parameters as in [4], we seem to have outperformed previous results, especially for a small window lengths. Looking at Figs. 1c and 1d, we can see that recurrent models strongly outperform the results from MLPs for small window sizes. For the ESN with one input, we can see that the area under the ROC curve increased from 0.75 (MLP) to 0.86 (ESN). Up to a window size of 5 samples, the unidirectional ESN outperformed the MLP. Afterwards, the MLP performs slightly better. Bidirectional ESNs again exceeded the performance of unidirectional models in almost any case. For larger windows, they perform similar as MLPs. However, we again note that bidirectional ESNs are not able to operate online, since this demands causality.

In summary, we see that all the models examined performed respectably. It can be seen that all neural networks performed better than the linear baselines, particularly for small window sizes. For short window configurations, we can see that the ESN models strongly outperformed the ridge regression and MLP models. For larger windows, the MLP performed better. Note that we optimized the hyper-parameters of the ESN towards online capabilities (window size 1) and did not changed it afterwards. Doing so could potentially improve the ESN performance and is something that will be investigated in future studies.

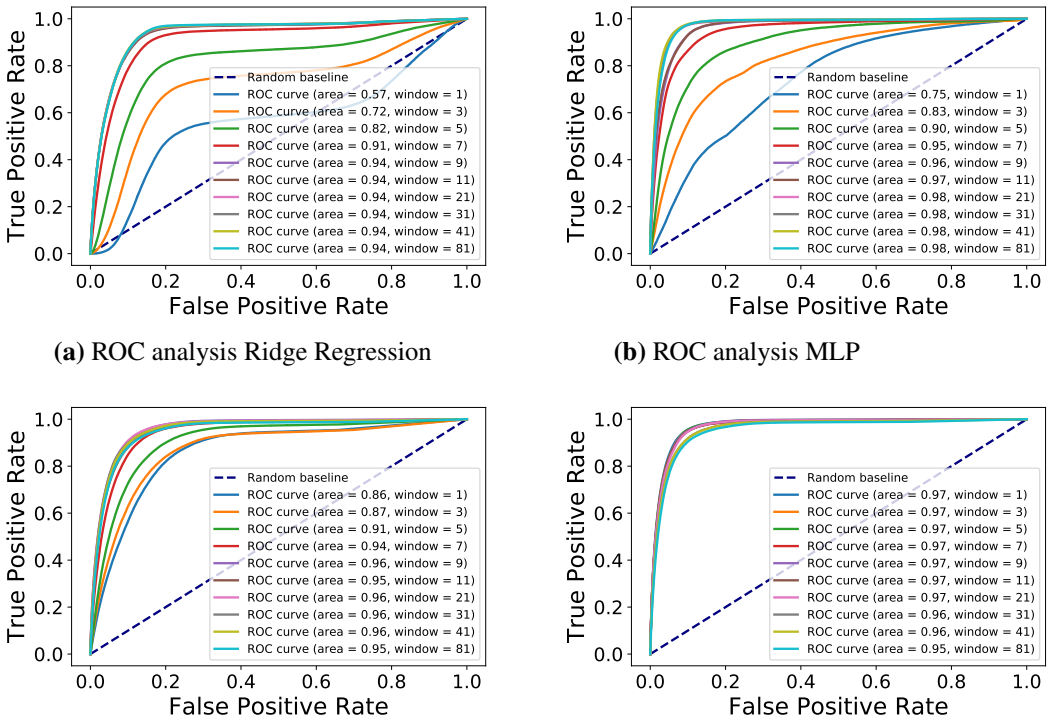
4.2 Noise robustness of different models

Next, we investigate the noise robustness of the best MLP and ESN models. As one key advantage of ESNs is their capability to internally model temporal context, we add the online models in this evaluation.

From Figure 2, we can see that the MLP (Figure 2a) in online configuration does not perform very well. Only in case of rather clean speech (up to an SNR of 20 dB) the model performs above chance level. The unidirectional ESN (Figure 2c) is much more robust against noise. Only for very noisy speech (SNR levels of -10 dB and -20 dB), the performance is very close to chance level.

5 Conclusions and outlook

We have compared and evaluated several models for GCI estimation from raw speech signals. Particularly, we have shown the impact of adding complexity to the model. Even a linear model with sufficient context information was able to generate respectable performance. We were able to improve on this baseline performance by adopting the MLP architecture described in [4].



(a) ROC analysis Ridge Regression

(b) ROC analysis MLP

(c) ROC analysis unidirectional ESN

(d) ROC analysis bidirectional ESN

Figure 1 – Receiver operating characteristic (ROC) analysis of different configurations: Ridge regression performed worst, especially in an online configuration. The MLP results were better. In case of short temporal windows as for an online configuration the unidirectional ESN outperformed the MLP.

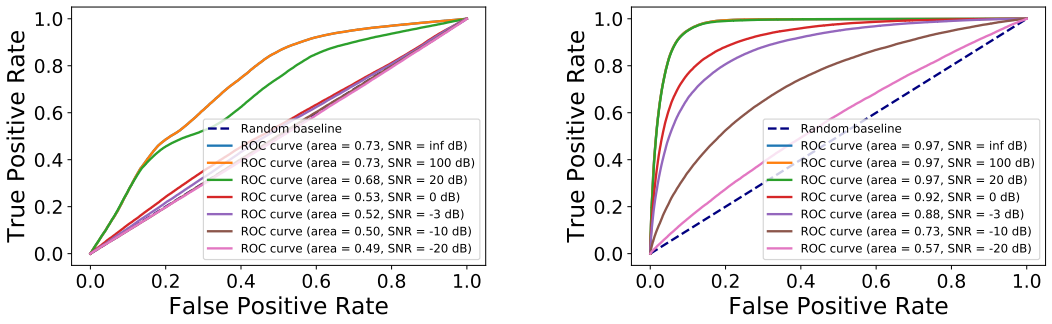
When provided with sufficient temporal contextual information, this MLP model was able to achieve very good results.

A current research trend are online models for various classification and regression tasks. In that case, the MLP performed worse, because it was unable to handle temporal context without explicit input. Thus, we replaced the MLP by a simple ESN. With this approach, we strongly improved the online capability for GCI estimation. Without changing hyper-parameters and after switching to bidirectional architectures, we achieved similar results as the MLP with temporal context.

In the future, we will expand the approach in various ways. The high input scaling factor suggests that an audio input signals should be either normalized or models need to be trained with augmented datasets including various maximum amplitudes. The ground truth could be improved.

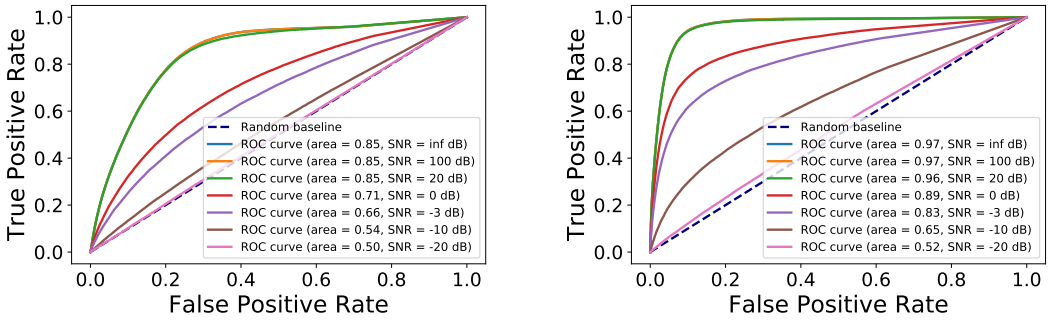
It would be interesting to extract the f_0 contour from the resulting GCI detection function and compare this to reference contours from standard pitch tracking algorithms implemented in Praat [13] or YIN [14]. Such algorithms still obtain very promising results, and the proposed models used here need to be compared to these standard references.

Complexity can be removed from the ESN model by utilizing smaller reservoirs. The ESN model can be pre-trained using the K-Means algorithm. Furthermore, it is interesting to generalize this approach towards a large variety of speakers, especially female speakers. Finally, to improve the capabilities of ESNs, it is to be noted that the binary target function is not very optimal for a regression training of the ESN. In this paper, we have set the samples surrounding GCI instants to 0.5. In the future, we will adapt the approach [15] that used a real glottal source model as the target function, which is not binary.



(a) ROC online MLP

(b) ROC analysis MLP 81 samples



(c) ROC online ESN

(d) ROC analysis bidirectional ESN 81 samples TODO

Figure 2 – Receiver operating characteristic (ROC) analysis of the best performing and causal MLP and ESN models for different SNR levels: The causal MLP is not very robust against noise. Although it was trained with an SNR of 0 dB, the performance at this specified SNR is already close to chance level. In case of the large window size (81 samples), no degradation up to an SNR of 20 dB is observed. With an SNR of -3 dB, the area under the ROC curve is still 0.88. Only for very noisy speech samples, the performance is no longer robust. The ESN in online settings performed better than the corresponding MLP. However, the performance for very low SNR levels (-10 dB and -20 dB) also here is close to chance level.

6 Acknowledgements

The parameter optimizations were performed on a Bull Cluster at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden.
This research was financed by Europäischer Sozialfonds (ESF) and the Free State of Saxony (Application number: 100327771).
We thank Adrian Fourcin (University College London, UK) for giving us permission to use the EGG dataset.

References

[1] RAJAEI, A., E. BARZEGAR, F. MOJIRI, and M. NILFOROUSH: *The occurrence of laryngeal penetration and aspiration in patients with glottal closure insufficiency*. *ISRN otolaryngology*, 2014, p. 587945, 2014. doi:10.1155/2014/587945.

[2] REDDY, M. G. and T. M. K. S. RAO: *Glottal closure instants detection from pathological acoustic speech signal using deep learning*. In *Proceedings of the Machine Learning for Health Workshop*. 2018.

[3] HOWARD, I.: *Speech fundamental period estimation using pattern classification*. Ph.D. thesis, University of London, 1991. doi:10.13140/RG.2.2.35011.81449.

[4] HOWARD, I. S.: *Speech fundamental period estimation using a neural network*. In R. BÖCK, I. SIEGERT, and A. WENDEMUTH (eds.), *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*, pp. 44–51. TUDpress, Dresden, 2020.

[5] MOULINES, E. and F. CHARPENTIER: *Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones*. *Speech Communication*, 9(5-6), pp. 453–467, 1990.

[6] STEINER, P., S. STONE, P. BIRKHOLZ, and A. JALALVAND: *Multipitch tracking in music signals using echo state networks*. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pp. 126–130. 2020. URL https://www.eurasip.org/Proceedings/Eusipco/Eusipco2020/pdfs/0000126.pdf.

[7] STEINER, P., S. STONE, and P. BIRKHOLZ: *Note onset detection using echo state networks*. In R. BÖCK, I. SIEGERT, and A. WENDEMUTH (eds.), *Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2020*, pp. 157–164. TUDpress, Dresden, 2020.

[8] JALALVAND, A., K. DEMUYNCK, W. D. NEVE, and J.-P. MARTENS: *On the application of reservoir computing networks for noisy image recognition*. *Neurocomputing*, 277, pp. 237 – 248, 2018. doi:https://doi.org/10.1016/j.neucom.2016.11.100. URL http://www.sciencedirect.com/science/article/pii/S0925231217314145. Hierarchical Extreme Learning Machines.

[9] JAEGER, H.: *The "echo state" approach to analysing and training recurrent neural networks*. Tech. Rep. GMD Report 148, German National Research Center for Information Technology, 2001. URL http://www.faculty.iu-bremen.de/hjaeger/pubs/EchoStatesTechRep.pdf.

[10] FAWCETT, T.: *An introduction to roc analysis*. *Pattern Recognition Letters*, 27(8), pp. 861 – 874, 2006. doi:https://doi.org/10.1016/j.patrec.2005.10.010. URL http://www.sciencedirect.com/science/article/pii/S016786550500303X. ROC Analysis in Pattern Recognition.

[11] MCFEE, B., C. RAFFEL, D. LIANG, D. P. ELLIS, M. MCVICAR, E. BATTENBERG, and O. NIETO: *librosa: Audio and Music Signal Analysis in Python*. In *Proceedings of the 14th Python in Science Conference (SciPy 2015)*, vol. 8, pp. 18–25. 2015.

[12] STEINER, P., A. JALALVAND, S. STONE, and P. BIRKHOLZ: *Feature engineering and stacked echo state networks for musical onset detection*. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 1–8. 2020.

[13] BOERSMA, P. and V. V. HEUVEN: *Speak and unspeak with praat*. *Glot International*, 5(9/10), pp. 341–347, 2001.

[14] CHEVEIGNÉ, A. D. and H. KAWAHARA: *Yin, a fundamental frequency estimator for speech and music*. *The Journal of the Acoustical Society of America*, 111(4), pp. 1917–1930, 2002.

[15] ARDAILLON, L. and A. ROEBEL: *Gci detection from raw speech using a fully-convolutional network*. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6739–6743. 2020. doi:10.1109/ICASSP40776.2020.9053089.