USING EXTREME GRADIENT BOOSTING TO DETECT GLOTTAL CLOSURE INSTANTS IN SPEECH SIGNAL

Jindřich Matoušek[1,2], *Daniel Tihelka*[2]

[1]Department of Cybernetics, [2]New Technology for the Information Society (NTIS)
Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Rep.

ABSTRACT

In this paper, we continue to investigate the use of classifiers for the automatic detection of glottal closure instants (GCIs) from the speech signal. We focus on extreme gradient boosting (XGB), a fast and powerful implementation of a gradient boosting algorithm. We show that XGB outperforms other classifiers, achieving GCI detection accuracy $F1 = 98.55\%$ and $AUC = 99.90\%$. The proposed XGB model is also shown to outperform other existing GCI detection algorithms on publicly available databases. Despite using much less training data, the performance of XGB is comparable to a deep convolutional neural network based approach, especially when it is tested on voices that were not included in the training data.

Index Terms— glottal closure instant (GCI), pitch mark, detection, classification, extreme gradient boosting

1. INTRODUCTION

Glottal closure instants (GCIs) (also called *pitch marks* or *epochs*) refer to peaks in *voiced parts* of the speech signal that correspond to the moment of glottal closure, a significant excitation of a vocal tract. The distance between two succeeding GCIs then corresponds to one vocal fold vibration cycle and can be represented in the time domain by a local *pitch period* value (T_0) or in the frequency domain by a local *fundamental frequency* value (F_0).

Precise detection of GCIs plays a key role in *pitch-synchronous* speech processing which is used in many speech-technology applications [1, 2, 3, 4]. Although GCIs can be reliably detected from a simultaneously recorded electroglottograph (EGG) signal (which measures glottal activity directly; thus, it is not burdened by modifications that happen to a flow of speech in the vocal tract – see Figure 1c), for the sake of simplicity, in many practical applications it is often desirable to detect GCIs directly from the speech signal only.

In our previous work [5, 6], we showed that GCI detection could be viewed as a two-class classification problem: whether or not a peak in a speech waveform represents a GCI [7, 8, 9]. This is quite a different approach compared to traditionally used algorithms which usually use expert knowledge and hand-crafted rules and thresholds to identify GCI candidates from local maxima of various speech representations (e.g. linear predictive coding like in DYPSA [10], YAGA [2] or [11], wavelet components [12], multiscale formalism (MMF) [13]) and/or from discontinuities or changes in signal energy (Hilbert envelope, Frobenius norm, zero-frequency resonator, or SEDREAMS [14]). Dynamic programming is then often used to refine the GCI candidates [10, 2, 15].

This research was supported by the Czech Science Foundation (GA CR), project No. GA19-19324S. The access to the MetaCentrum clusters provided under the programme LM2015042 is highly appreciated.

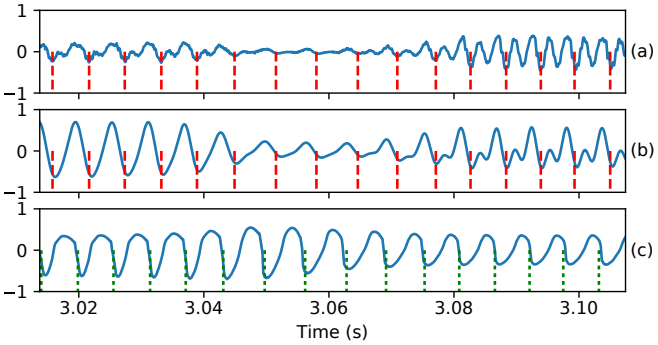

Fig. 1. Example of a speech signal (a), the corresponding low-pass filtered signal (b), and EGG signal (c). GCIs are marked by red dashed (speech signal) and green dotted (EGG signal) lines.

The advantage of the classification-based method is that once a training dataset is available, classifier parameters are set up automatically without manual tuning. We showed the classification-based GCI detection was able to perform very well and consistently outperformed traditionally used algorithms on several test datasets.

In this paper, we continue to investigate the use of classifiers for GCI detection [5, 6]. We focus on *gradient boosting decision tree* algorithm, a powerful "non-deep" learning technique for building predictive models. More specifically, *extreme gradient boosting* (XGB) [16], an implementation of gradient boosted decision trees designed for speed and performance that dominates many Kaggle competitions, is researched here within the GCI detection problem.

2. EXPERIMENTAL DATA

Experiments were performed on clean 16kHz-sampled speech recordings available at our workplace (hereafter referred to as UWB) [17, 6]. The recordings were primarily intended for speech synthesis. We used 63 utterances (≈9 minutes of speech) for the development of the proposed XGB classifier, and 20 test utterances (≈3 minutes of speech) were held out for an unbiased comparison with other methods. The set of utterances was the same as in [17] – it comprised various Czech (male and female), Slovak (female), German (male), US English (male), and French (female) voices. All voices were part of both the development and test datasets. Reference GCIs produced by a human expert (using both speech and EGG signals) were available for each utterance and were synchronized with the corresponding minimum negative sample in the speech signal.

Speech waveforms were processed in the same way as described in [6]. They were mastered to have equal loudness, low-pass filtered by a zero-phase Equiripple-designed filter with 0.5 dB ripple in the

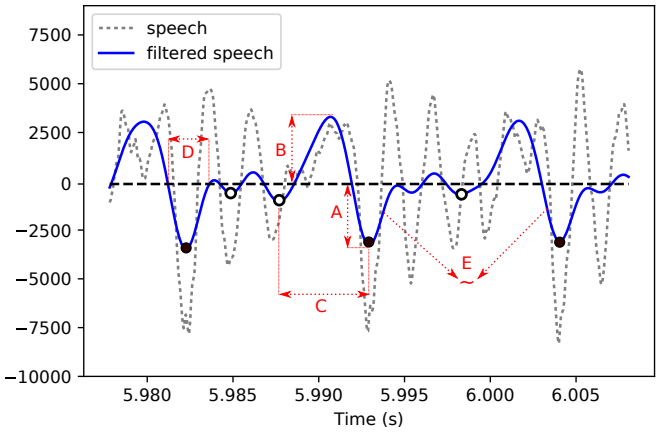

Fig. 2. Illustration of feature extraction: amplitude of a negative peak (A, negAmp), amplitude of a positive peak (B, posAmp), difference between two negative peaks (C, timeDiff), width of a negative peak (D, width), correlation between waveforms of two negative peaks (E, corr). GCI candidates are marked by ○, true GCIs by ●.

pass band, 60 dB attenuation in the stop band, and with the cutoff frequency of 800 Hz to reduce the high-frequency structure in the speech signal (see Figure 1b). The signals were then zero-crossed to identify peaks (both of the negative and positive polarity) that are used for feature extraction in further processing. Since the polarity of speech signals was shown to have an important impact on the performance of a GCI detector [18, 19], all speech signals were switched to have the negative polarity, and only the negative peaks were taken as the candidates for the GCI placement. For the purpose of training and testing, the location of each reference GCI was mapped to a corresponding negative peak in the filtered signal. There were 73205 and 20338 candidate peaks in the development and test datasets respectively (marked by both ○ and ● in Figure 2), 39931 and 10807 of them corresponded to true GCIs (marked by ● only).

The baseline features used are illustrated in Figure 2. Inspired by [9], the features were associated with negative peaks in the low-pass filtered speech waveforms. Each peak is described by a set of local descriptors reflecting the position and shape of other 3 neighboring peaks [5]. Thus, only 32 features were used in total: the amplitudes of the given negative peak and 6 neighboring (3 prior and 3 subsequent) negative peaks (7 features, denoted as A in Figure 2), amplitudes of 6 neighboring positive peaks (6, B), the time difference between the given negative peak and each of the neighboring negative peaks (6, C), the width of the given negative peak (a distance between two zero-crossings) and each of the neighboring negative peaks (7, D), the correlation of the waveform around the given negative peak and the waveforms around each of the neighboring negative peaks (6, E).

3. EXTREME GRADIENT BOOSTING

In principle, gradient boosting algorithm uses an *ensemble* technique called *boosting* to add new models (decision trees) in order to correct errors made by existing models. Boosting is repeated until no further improvements can be made. *Gradient boosting* is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. *Gradient descent algorithm* is used to minimize the loss when adding new models. We used a fast and powerful implementation of gradient boosting, *extreme gradient boosting* (XGB) [16]. Important hyper-parameters of the XGB model are shown in Table 1.

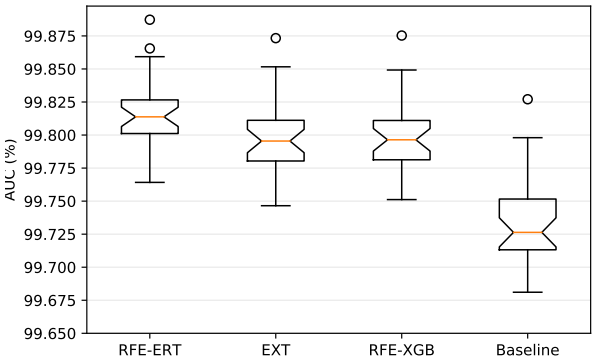

Fig. 3. Box-and-whisker plot comparison (with whiskers corresponding to 1.5 IQR) of different feature sets on the validation dataset with respect to AUC score.

Table 1. Important hyper-parameters of the XGB model and their default and optimized values. For the explanation of the hyper-parameters, see e.g. [16].

Hyper-parameter	Default	Optimized
number of trees	100	1068
boosting learning rate (η)	0.1	0.1
maximum tree depth	3	7
minimum child weight	1	1
min. loss reduction required to make a split (γ)	0	0
subsample ratio of the training instance	1	0.9
subsample ratio of columns for tree construction	1	0.65
subsample ratio of columns for each split	1	0.6
L1 regularization term on weights (α)	0	1e-08
L2 regularization term on weights (λ)	1	1

To evaluate the performance of the proposed XGB model and to compare it to other classifiers, standard classification measures like *recall* (R), *precision* (P), $F1$, and area under the receiver operating characteristic curve (AUC) were utilized. *Scikit-learn* [20] and *XGBoost* [16] toolkits were employed to train and evaluate the proposed models.

3.1. Feature engineering

To find the best feature set, we extended the baseline feature set (32 features) described in Section 2 with *acoustic features* (zero-crossing rate (ZCR), log energy, harmonic-to-noise ratio (HNR), voiced/unvoiced, peak ratio to 6 neighboring peaks), *spectral features* (spectral centroid, spectral bandwidth, and spectral roll-off), and *mel-frequency cepstral coefficients* (MFCCs). All features were calculated from 10ms-long speech segments extracted around every peak candidate [6]. There were 58 features in the extended feature set (EXT).

We also applied a feature selection algorithm (*recursive feature elimination*, RFE [21]) to select important features automatically. Starting from the full feature set EXT, the RFE algorithm recursively prunes out the least important features until the desired number of features is reached. The desired number of features was selected by the 10-fold cross-validation technique. The feature importance was assigned by an external estimator – *extremely randomized trees* (ERT) [22], which yielded the best performance in [6], and also extreme gradient boosting (XGB). The optimal feature set selected by RFE-ERT consisted of 37 features and by RFE-XGB of 39 features.

Since decision tree based techniques are generally known to be invariant to data scaling, no data scaling/preprocessing was applied

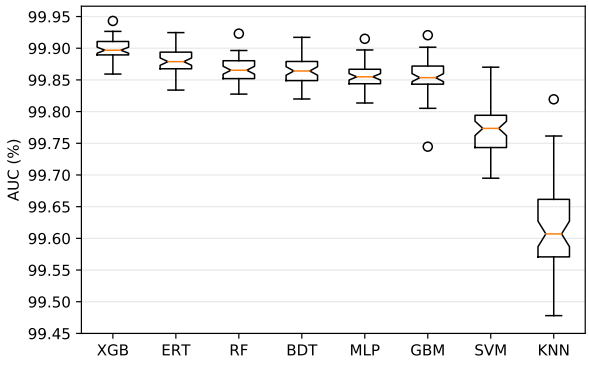

Fig. 4. Box-and-whisker plot comparison (with whiskers corresponding to 1.5 IQR) of classifiers' GCI detection performance on the validation dataset with respect to AUC score.

Table 2. Comparison of classifiers' GCI detection performance on the validation dataset in terms of recall (R), precision (P), $F1$, and AUC score.

Model	R (%)	P (%)	$F1$ (%)	AUC (%)
XGB	**98.39**	98.71	**98.55**	**99.90**
ERT	98.21	**98.72**	98.46	99.88
BDT	98.06	98.65	98.35	99.86
GBM	98.27	98.52	98.40	99.86
RF	98.03	98.69	98.36	99.87
MLP	98.26	98.31	98.28	99.86
SVM	98.21	98.56	98.38	99.77
KNN	98.05	98.51	98.28	99.62

when developing the XGB model. The XGB classifier with the default hyper-parameters shown in Table 1 [16] was trained and evaluated on the development data described by the different feature sets using the repeated 10-fold cross-validation strategy (with the number of repeats being 3). The comparison of the different feature sets in Figure 3 indicate that the RFE-ERT algorithm yields the best results.

3.2. Model tuning

Extensive XGB model hyper-parameter tuning using grid search with 10-fold cross-validation was conducted on the development dataset. For the hyper-parameter optimization, AUC measure was used. The RFE-ERT based feature set described in Section 3.1 was utilized. The optimized hyper-parameter values are shown in Table 1.

3.3. Comparison with other classifiers

We also trained, tuned and evaluated a number of other classifiers [6]. Some of them were *decision-tree based ensemble models* similar to XGB: bagged decision trees (BDT) [23], random forests (RF) [24], extremely randomized trees (ERT) [22]), and gradient boosting machines (GBM) [25]. The other ones were *non-linear* classifiers like support vector machines (SVM) with a Gaussian radial basis function (RBF) kernel, multilayer perceptron (MLP) and k-nearest neighbors (KNN). We used scikit-learn implementations of these models [20].

Repeated 10-fold cross-validation (with the number of repeats being 5) was employed to train and compare the classifiers on the development set. As can be seen in Figure 4 and Table 2, the proposed XGB classifier yields the best performance. Results in Figure 4 can be interpreted that XGB is significantly better than all other classifiers.

4. COMPARISON WITH OTHER METHODS

In the previous section, the proposed model was evaluated in a standard classification-manner, i.e., how good the classifier is *both in classifying peaks that correspond to true GCIs and, at the same time, in classifying peaks that do not represent GCIs*. Now, however, we will look at the comparison of the GCI detection with some other available detection algorithms.

4.1. Performance measures

The most common way to assess the performance of GCI detection techniques is to compare *locations of the detected and reference GCIs*. The widely used measures typically concern the *reliability* and *accuracy* of the GCI detection algorithms [10]. The former includes the percentage of glottal closures for which exactly one GCI is detected (*identification rate*, IDR), the percentage of glottal closures for which no GCI is detected (*miss rate*, MR), and the percentage of glottal closures for which more than one GCI is detected (*false alarm rate*, FAR). The latter includes the percentage of detections with the identification error $\zeta \leq 0.25$ ms (*accuracy to* ± 0.25 *ms*, A25) and standard deviation of the identification error ζ (*identification accuracy*, IDA).

4.2. Compared methods

We compared the proposed extreme gradient boosting model with four existing state-of-the-art GCI detection methods:

- *Speech Event Detection using the Residual Excitation And a Mean-based Signal* (SEDREAMS) [14] (available in the COVAREP repository [26, 27], v1.4.1), shown in [1] to provide the best of performances compared to other methods;
- fast GCI detection based on *Microcanonical Multiscale Formalism* (MMF) [13] (available in [28]);
- *Dynamic Programming Projected Phase-Slope Algorithm* (DYPSA) [10] available in the VOICEBOX toolbox [29];
- Google's *Robust Epoch And Pitch EstimatoR* (REAPER) [30].

We used the implementations available online; no modifications of the algorithms were made. Since all algorithms (except REAPER) estimate GCIs also during unvoiced segments, authors recommend filtering the detected GCIs by the output of a separate voiced/unvoiced detector. We applied an F_0 contour estimated by the REAPER algorithm for this purpose. There is no need to apply such a postprocessing on GCIs detected by the proposed classification-based approach since the voiced/unvoiced pattern was included directly in the feature set (see Section 3.1). To obtain consistent results for all methods, the detected GCIs were shifted towards the neighboring minimum negative sample in the speech signal [6].

4.3. Test datasets

Firstly, the evaluation was carried out on the UWB test dataset (≈3 minutes of speech) described in Section 2. GCIs produced by a human expert were used as reference GCIs.

Secondly, two voices, a US male (BDL) and a US female (SLT) from the CMU ARCTIC databases intended for unit selection speech synthesis [31, 32] were used as a test material. Each voice consists of 1132 phonetically balanced utterances of a total duration ≈54 minutes per voice. Additionally, KED TIMIT database [32] comprising 453 phonetically balanced utterances (≈20 min.) of a US male speaker was also used for testing. All these datasets comprise

Table 3. Summary of the performance of the GCI detection algorithms for the four datasets.

Dataset	Method	IDR (%)	MR (%)	FAR (%)	IDA (ms)	A25 (%)
UWB	XGB	**96.63**	**2.20**	**1.17**	**0.23**	**98.71**
	SEDREAMS	93.14	3.99	2.87	0.29	98.09
	MMF	85.09	11.42	3.48	0.47	97.86
	DYPSA	89.62	6.26	4.12	0.37	98.07
	REAPER	92.62	5.60	1.78	0.25	98.31
BDL	XGB	**93.85**	**2.37**	3.78	0.45	95.74
	SEDREAMS	91.82	3.02	5.16	**0.44**	**97.37**
	MMF	89.49	4.53	5.98	0.57	96.23
	DYPSA	88.95	4.32	6.73	0.56	96.81
	REAPER	93.24	4.39	**2.38**	0.59	97.02
SLT	XGB	**96.05**	**0.57**	3.38	0.21	98.69
	SEDREAMS	94.67	1.12	4.21	**0.18**	**99.61**
	MMF	92.48	5.24	**2.28**	0.41	98.89
	DYPSA	93.23	2.88	3.89	0.31	99.39
	REAPER	95.48	1.71	2.82	0.21	99.23
KED	XGB	**96.69**	**1.29**	3.02	**0.26**	**99.55**
	SEDREAMS	92.31	6.03	1.66	0.29	99.04
	MMF	90.24	7.04	2.72	0.37	98.79
	DYPSA	90.29	7.05	2.66	0.31	99.16
	REAPER	91.04	8.18	**0.78**	0.27	99.45

Table 4. Comparison of the performance of XGB and CNN based classifiers.

Dataset	Method	IDR (%)	MR (%)	FAR (%)	IDA (ms)	A25 (%)
SLT	XGB	96.05	**0.57**	3.38	0.21	98.69
	CNN-MIX2	94.87	4.51	0.62	**0.03**	99.46
	CNN-MIX4	**97.51**	2.27	**0.22**	**0.03**	**99.47**
KED	XGB	**96.69**	**1.29**	3.02	0.26	**99.55**
	CNN-MIX2	91.51	6.87	1.62	**0.02**	96.98
	CNN-MIX4	94.61	5.12	**0.26**	**0.02**	98.31

clean speech. Since there are no hand-crafted GCIs available for these datasets, GCIs detected from contemporaneous EGG recordings by the Multi-Phase Algorithm (MPA) [33] (again shifted towards the neighboring minimum negative sample in the speech signal) were used as the reference GCIs (the reference GCIs and other data relevant to the described experiments are available online [34]). Original speech signals were downsampled to 16 kHz. It is important to mention that no voice from these datasets was part of the training dataset used to train the proposed XGB classifier.

4.4. Results

The results in Table 3 show that the proposed XGB model performs very well for all tested datasets[1]. It excels in terms of *reliability*, especially with respect to the identification (IDR) and miss (MR) rates. As for the *accuracy*, XGB performed very well as it achieved, together with the SEDREAMS algorithm, the highest identification accuracy (IDA) and yielded the smallest number of timing errors higher than 0.25 ms (A25).

We also compared the XGB model to another popular classification-based method – *deep convolutional neural network* (CNN) proposed for GCI detection by Yang et al. [35]. Much more training data is required for CNN: 900 utterances from BDL and JMK (another male voice from the ARCTIC repository [32]) datasets were used to develop a CNN model (CNN-MIX2) and even 1500 utterances from BDL, JMK, SLT, and KED datasets were used to develop another CNN model (CNN-MIX4). Although the results in Table 4 are not directly comparable (the results of CNN models were evaluated on a subset of 500 SLT and 300 KED utterances, and the reference gold-truth GCIs were obtained in a different way – see [35] for more detail), XGB developed on much less data (63 utterances only) generally outperforms CNN-MIX2 on both test datasets in terms of reliability and, for the KED dataset, XGB outperforms also CNN-MIX4. Note that SLT and KED voices were also part of CNN-MIX4 training dataset; this was not the case of XGB and CNN-MIX2 models.

[1] A possible explanation of lower performance metrics (cf. e.g. [1, 10]) is the use of different reference GCIs, a different strategy of GCI filtering in unvoiced segments, and perhaps also a different implementation of GCI computation evaluation (also available in [34]).

5. CONCLUSIONS

In this paper, we followed up on our previous work concerning the use of classifiers to detect GCIs in the speech signal. We showed that the extreme gradient boosting classifier performs best when the baseline set of features is extended with other acoustic, spectral, and MFCC-based features, and the final set of features is selected automatically using the recursive feature elimination technique. The proposed XGB classifier was shown to outperform other classifiers, achieving GCI detection accuracy $F1 = 98.55\%$ and $AUC = 99.90\%$. The XGB classifier also yielded the best results when compared to other existing state-of-the-art methods on several test datasets. Despite using much less training data, it also performed well in comparison with a deep convolutional neural network, especially when tested on voices that were not included in the training data.

6. REFERENCES

[1] T. Drugman, M. Thomas, J. Gudnason, P. Naylor, and T. Dutoit, "Detection of glottal closure instants from speech signals: A quantitative review," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 994–1006, mar 2012.

[2] M. R. P. Thomas, J. Gudnason, and P. A. Naylor, "Estimation of glottal closing and opening instants in voiced speech using the YAGA algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 82–91, jan 2012.

[3] T. Drugman, P. Alku, A. Alwan, and B. Yegnanarayana, "Glottal source processing: From analysis to applications," *Computer Speech and Language*, vol. 28, no. 5, pp. 1117–1138, 2014.

[4] J. Matoušek and J. Romportl, "Automatic pitch-synchronous phonetic segmentation," in *INTERSPEECH*, Brisbane, Australia, 2008, pp. 1626–1629.

[5] J. Matoušek and D. Tihelka, "Classification-based detection of glottal closure instants from speech signals," in *INTERSPEECH*, Stockholm, Sweden, 2017, pp. 3053–3057.

[6] ——, "Glottal closure instant detection from speech signal using voting classifier and recursive feature elimination," in *INTERSPEECH*, Hyderabad, India, 2018, pp. 2112–2116.

[7] I. S. Howard and M. A. Huckvale, "Speech fundamental period estimation using a trainable pattern classifier," in *SPEECH'88: 7th FASE Symposium*, Edinburgh, UK, 1988.

[8] J. R. Walliker and I. S. Howard, "Real-time portable multi-layer perceptron voice fundamental-period extractor for hearing aids and cochlear implants," *Speech Communication*, vol. 9, no. 1, pp. 63–72, 1990.

[9] E. Barnard, R. A. Cole, M. P. Vea, and F. A. Alleva, "Pitch detection with a neural-net classifier," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 298–307, 1991.

[10] P. A. Naylor, A. Kounoudes, J. Gudnason, and M. Brookes, "Estimation of glottal closure instants in voiced speech using the DYPSA algorithm," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 34–43, 2007.

[11] A. P. Prathosh, T. V. Ananthapadmanabha, and A. G. Ramakrishnan, "Epoch extraction based on integrated linear prediction residual using plosion index," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 21, no. 12, pp. 2471–2480, 2013.

[12] V. N. Tuan and C. D'Alessandro, "Robust glottal closure detection using the wavelet transform," in *EUROSPEECH*, Budapest, Hungary, 1999, pp. 2805–2808.

[13] V. Khanagha, K. Daoudi, and H. M. Yahia, "Detection of glottal closure instants based on the microcanonical multiscale formalism," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1941–1950, 2014.

[14] T. Drugman and T. Dutoit, "Glottal closure and opening instant detection from speech signals," in *INTERSPEECH*, Brighton, Great Britain, 2009, pp. 2891–2894.

[15] P. Sujith, A. P. Prathosh, R. A. G., and P. K. Ghosh, "An error correction scheme for GCI detection algorithms using pitch smoothness criterion," in *INTERSPEECH*, Dresden, Germany, 2015, pp. 3284–3288.

[16] T. Chen and C. Guestrin, "XGBoost: Reliable Large-scale Tree Boosting System," in *Conference on Knowledge Discovery and Data Mining*, 2016.

[17] M. Legát, J. Matoušek, and D. Tihelka, "On the detection of pitch marks using a robust multi-phase algorithm," *Speech Communication*, vol. 53, no. 4, pp. 552–566, 2011.

[18] M. Legát, D. Tihelka, and J. Matoušek, "Pitch marks at peaks or valleys?" in *Text, Speech and Dialogue*, ser. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2007, vol. 4629, pp. 502–507.

[19] T. Drugman, "Residual excitation skewness for automatic speech polarity detection," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 387–390, 2013.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. M. B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perror, and É. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1/3, pp. 389–422, 2002.

[22] P. Geurts and D. E. L. Wehenkel, "Extremely Randomized Trees," *Machine Leaming*, vol. 63, no. 1, pp. 3–42, 2006.

[23] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[24] ——, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[25] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[26] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer, "COVAREP - A collaborative voice analysis repository for speech technologies," in *IEEE International Conference on Acoustics Speech and Signal Processing*, Florence, Italy, 2014, pp. 960–964.

[27] "A Cooperative voice analysis repository for speech technologies." [Online]. Available: https://github.com/covarep/covarep

[28] "Matlab codes for Glottal Closure Instants (GCI) detection." [Online]. Available: https://geostat.bordeaux.inria.fr/index.php/downloads.html

[29] "VOICEBOX: Speech Processing Toolbox for MATLAB." [Online]. Available: http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[30] "REAPER: Robust Epoch And Pitch EstimatoR." [Online]. Available: https://github.com/google/REAPER

[31] J. Kominek and A. W. Black, "The CMU ARCTIC speech databases," in *Speech Synthesis Workshop*, Pittsburgh, USA, 2004, pp. 223–224.

[32] "FestVox Speech Synthesis Databases." [Online]. Available: http://festvox.org/dbs/index.html

[33] M. Legát, J. Matoušek, and D. Tihelka, "A robust multi-phase pitch-mark detection algorithm," in *INTERSPEECH*, vol. 1, Antwerp, Belgium, 2007, pp. 1641–1644.

[34] "Data used for extreme gradient boosting based glottal closure instant detection." [Online]. Available: https://github.com/ARTIC-TTS-experiments/2019-ICASSP

[35] S. Yang, Z. Wu, B. Shen, and H. Meng, "Detection of glottal closure instants from speech signals: a convolutional neural network based method," in *INTERSPEECH*, Hyderabad, India, 2018, pp. 317–321.