CLASSIFICATION OF BREATH AND SNORE SOUNDS USING AUDIO DATA RECORDED WITH SMARTPHONES IN THE HOME ENVIRONMENT

Tim Fischer, Johannes Schneider, Wilhelm Stork

FZI Research Center for Information Technology
Dept. of Embedded Systems and Sensors Engineering
Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany

Tim.Fischer@fzi.de, Johannes.Schneider@fzi.de, wilhelm.stork@kit.edu

ABSTRACT

In this paper, classification between snore-inhale (SI), snore-exhale (SE), breathe-inhale (BI), breathe-exhale (BE) and noise (NS) sounds is performed. The database is obtained from 7 subjects, who recorded whole night audio data in their private home environments with their own smartphones. Preprocessing is done by a modification of an adaptive noise suppression method [1]. The classification system consists of 5 binary RobustBoost classifiers (RBs) [2] applying the one-vs.-rest strategy and an artificial neural network (ANN) for voting on the outputs. ReliefF and Sequential Forward Selection (SFS) extract a 5-dimensional feature vector, consisting of psychoacoustic features from time and spectral domain. Sensitivity (Se) and specificity (Sp) in percent on a preprocessed (i.e. the signal contains only sound activity segments), representative $1\,\mathrm{h}\,20\,\mathrm{min}$ dataset are:
$Se_{SI,SE,BI,BE,NS} = \{80.91,\ 80.01\,34,12,\ 66.45,\ 29.53\}$
$Sp_{SI,SE,BI,BE,NS} = \{83.56,\ 91.70,\ 90.53,\ 83.32,\ 93.51\}$

Index Terms— Breath sounds detection, audio signal processing, class noise, multiclassification systems, psychoacoustics

1. INTRODUCTION

Polysomnography is an expensive and inconvenient method, which is in addition very costly and not always necessary. Furthermore the patient has to spend the night in an unfamiliar environment, which affects sleep duration and sleep efficiency [3]. As the number of smartphones is rising worldwide [4], a simple and reliable technology for the detection of breath sounds saves money for the health system in terms of an early warning system (Obstructive Sleepapnea, Cheyne-Stokes respiration etc.).
Current apps try to create a "snore score", by counting and measuring the intensity of SI sounds. As all these apps are not open source, an objective result about their quality is difficult to obtain. However, in [5] 3 apps with the *best* performance on snoring detection are experimentally evaluated in terms of SI with disturbing noises. The apps detected the 600 SI-sounds with a variance of σ^2 =3272.3.
More academic approaches for detecting snore sounds (SI, SE vs. Rest) are performed in [6], [7], [1] (SI vs. Rest) and [8]. Roughly, results are based on frequency band energies (FBE) with an unsupervised k-Means clustering [8], Mel Frequency Cepstral Coefficients (MFCCs) with a Hidden Markov Model (HMM) [7], and diverse features from time- spectral- and cepstral domain using an Ada-Boost classifier [1]. Se and Sp are about $95\,\%$ with condenser microphones in soundproof environments. Also detecting snoring (SI, SE vs. NS, BI, BE) but with smartphone data, [9] and [10] obtain an accuracy of $84.35\,\%$ and $95.07\,\%$ using FBE, Zero Crossing Rate, Linear Predictive Coding (LPC) and a K-Nearest-Neighbour-Classifier or formant analysis and a quadratic classifier.
The detection of snoring (S), breathing (B) and noise (N) (SI, SE vs. BI, BE vs. NS) is performed in [11] and [12], using condenser microphones. With MFCCs and HMM, [11] achieves the following accuracies (Acc): $Acc_S = 89\,\%$, $Acc_B = 73\,\%$, $Acc_N = 69\,\%$. In [12], Largest Lyapunov Exponents and Entropy are used with a Multiclass Support Vector Machine (M-SVM) and an Adaptive Neuro Fuzzy Inference System (ANFIS) achieving the results: (ANFIS) $Se_S = 73\,\%$, $Se_B = 53.62\,\%$, $PPV_S = 87.91\,\%$, $PPV_B = 50.8\,\%$; (M-SVM) $Se_S = 87.58\,\%$, Se_B=67.8 %, $PPV_S = 85.57\,\%$, $PPV_B = 60.32\,\%$.
To the knowledge of the authors, yet there has not been done research on classifying audio signals in SI-, SE-, BI-, BE- and NS-sounds with real-environment audio data from different smartphones. This study is performed within Matlab (2015a).

2. MATERIALS AND METHODS

2.1. Sound Database

The audio data for developing and testing a classification system, is gathered by an observational study of 7 subjects. All subjects use their own smartphone in their home environment. Full night audio recordings of $139\,\mathrm{h}\,40\,\mathrm{min}$ were collected.

Table 1. Database

Device	NS	SI	SE	BI	BE
1	32495	15205	0	601	11508
	54.33 %	25.24 %	0.00 %	1.00 %	19.24 %
2	33166	10081	289	3139	1325
	69.10 %	21.00 %	0.60 %	6.54 %	2.76 %
3	43505	0	705	1656	2134
	90.64 %	1.47 %	0.00 %	3.45 %	4.45 %
4	3721	1910	0	0	0
	66.08 %	33.92 %	0.00 %	0.00 %	0.00 %
5	41905	6095	0	0	0
	87.30 %	12.70 %	0.00 %	0.00 %	0.00 %
6	112798	0	0	0	13205
	89.52 %	0.00 %	0.00 %	0.00 %	10.48 %
7	33951	599	7662	4837	951
	70.73 %	1.25 %	15.96 %	10.08 %	1.98 %
Overall	301541	33890	8656	10233	29123
	(106502)	(33044)	(6731)	(7810)	(22911)
	78.64 %	10.14 %	2.23 %	2.67 %	7.60 %
	(60.17 %)	(18.67 %)	(3.8 %)	(4.41 %)	(12.94 %)

Devices: 1 Apple iPhone 5, 2 Apple iPhone 4s, 3 Apple iPhone 5, 4 Sony Xperia Z3, 5 Samsung Galaxy S5 LTE+, 6 Samsung Galaxy Alpha, 7 HTC One M8.
Terms in brackets refer to the data after preprocessing.

For privacy reasons, the exact position of the subjects' smartphones remains unknown. The sample rate is set to $16\,\mathrm{kHz}$ and the bit depth to $16\,\mathrm{Bit}$. The data is split into sections, each approx. $10\,\mathrm{min}$ long. A representative selection of 122 sections are manually labelled, based on the human perception. The feature selection and the evaluation of this data requires too much computational cost, during the process of development. Therefore, an extract of the labelled data is used, as presented in table 1.

2.2. Preprocessing

Preprocessing discards segments with no information, reduces noise and performs an event detection.
The first step, is to reduce quantization noise of the oversampled $16\,\mathrm{kHz}$ signal, to improve the SNR and achieve a higher independence on the built-in A/D converter of the smartphone. Applying a high pass filter with a cut off frequency of $8\,\mathrm{kHz}$ and downsampling the signal by a factor of 2, results in a decimated signal without loss of information in the relevant spectrum [13]. Furthermore, computational cost is saved by discarding half of the data.
To achieve quasi-stationary conditions of the short-time Fourier analysis, the audio signal is split into frames of $25\,\mathrm{ms}$ with $50\,\%$ overlap [14] and multiplied by a hamming window, to minimize the leakage effect.
Subsequently, an adaptive noise suppression based on spectral suppression process with the Wiener-filter is performed ([15] in [1]). In other words, the spectral-distance of each frame is compared to a noise-template. If it passes a threshold, the noise-template is updated with the spectral components of the current frame. The spectral components k of each frame l are weighted by the equation 1 [1]. SNR_{Prio} is obtained by using the "Decision Directed Method" as in [15].

$$G(k,l) = \max\left(\frac{SNR_{Prio}(k,l)}{SNR_{Prio}(k,l)+1}, -25\,\mathrm{dB}\right) \quad (1)$$

All indices of the estimated noise-frames $n(l)$, are stored for subsequent event-detection and filter methods (see figure 1). Successive frames $n(l)$ form a noise-segment $N_i(\mathbf{L})$, where i indicates the segment number. When retransforming the signal from spectral into time domain, all $n(l)$ are stored in a circular buffer *noiseBuff*, with a memory size of $20\,\mathrm{s}$. The *noiseBuff* is then used to create a dynamic energy threshold with the quantile operator Q, as shown in equation 2.

$$EnThresh = Q_{noiseBuff}(0.93) \quad (2)$$

The next step is to discard segments, which contain no information with respect to the classes NS, SI, SE, BI, BE. This is performed by comparing the corresponding $N_i(\mathbf{L})$ with the minimum distance of two events, which is set to $0.1\,\mathrm{s}$. This makes sure, that no intra-event information is lost. The soundactive-segments $S_i(\mathbf{L})$, have to be longer than the minimum event length of $0.2\,\mathrm{s}$ and the median energy value requires to be higher than *EnThresh* (see eq. 2), otherwise they are set to 0. This process is shown in figure 1. The resulting dataset processed in this way, can be seen in the last row of table 1. Simultaneously the normalized energies of the soundactive segments $S_i(\mathbf{L})$ (see 3), are compared to the dynamic threshold *LowEn* (see eq. 4). Points of intersection form the segmentation edges for the event detection (see figure 1). Detected events are further checked with respect to their length and distance and consequently either merged, separated or discarded.

$$E_N(S(\mathbf{L})) = \left(\frac{s(l)^2}{\max(S(\mathbf{L})^2)}\right) \quad \mathrm{l} = 1...\mathrm{length}(\mathbf{L}) \quad (3)$$

$$LowEn = Q_{E_N(S(\mathbf{L}))}(0.93)\cdot 0.05 \quad (4)$$

2.3. Feature Selection

Feature Selection is performed by first using the ReliefF algorithm [16], followed by Sequential Forward Selection (SFS) [17].
As 899 features from time-, spectral- and cepstral domain are implemented, an automated process of selecting a subset of relevant features is necessary. Testing classifiers with many features requires too high computational cost, thus a pre-filtering of the features is done using the ReliefF algorithm [16]. ReliefF is a filter type method and selects features regardless of the classifier. As the performance of the ReliefF algorithm is dependent on the K nearest neighbours per class, the selection is processed for $K = \{10, 30, 50, 70, 100, 200, 300, 400, 500\}$. The selected

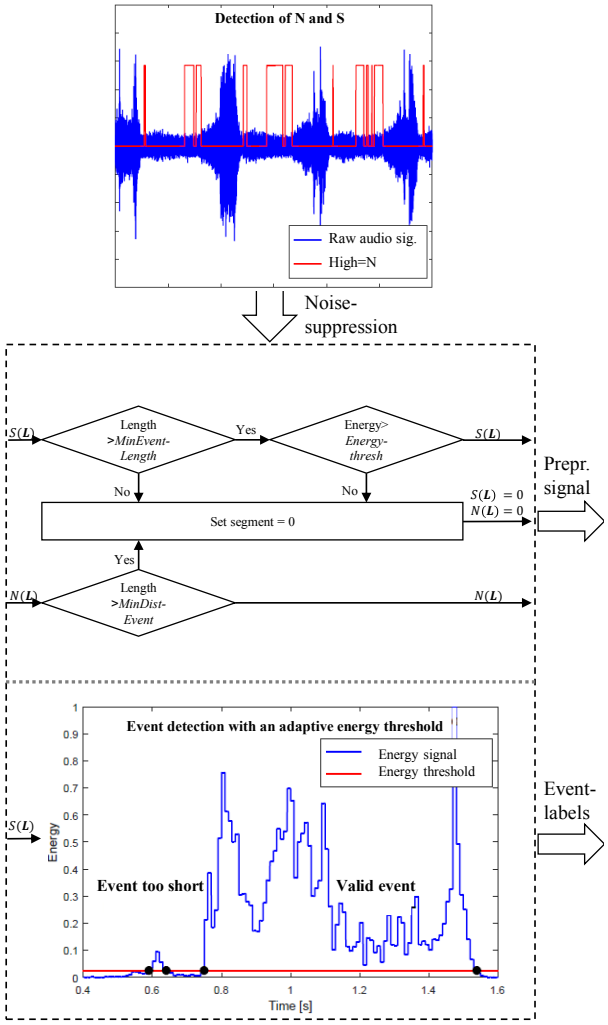

Fig. 1. Filtering segments with no information

features are chosen by creating the intersection of the 100 most relevant features for every K. Furthermore the 20 most relevant features of every K are added to the subset, unless already included. This results in a reduction from 899 to 85 features. These features are highly correlated, because ReliefF does not take this into account. Now SFS is applied on these 85 features using a RB (see 2.5) for {SI, SE} with 300 decision trees and an error goal of 20 %. Improvement in misclassification rate of at least 0.4 % is used for stopping criteria of the SFS algorithm. Applying first ReliefF and then SFS, 5 features are selected as described in table 2 and presented in their correlation matrix in figure 2. A more detailed description can be found in section 2.4. Due to complexity reduction, the classes {SI, SE} are merged. Adding another feature subset for {BI, BE} or NS, did not lead to satisfactory results, in regard to the computational time required.

Table 2. Features which are selected by the ReliefF and SFS method

No.	Name	Resolution
F1	First MFCC	25 ms
F2	Third LPC Cepstrum Coeff.	25 ms
F3	Modified Mel-Cepstability (MMC)	Event
F4	Change in specific loudness (CSL)	300 ms
F5	Pos. / neg. amplitude ratio	25 ms

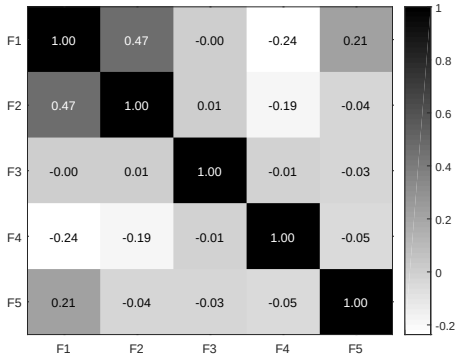

Fig. 2. Correlation matrix of the selected features

2.4. Selected Features

Short explanation of the features in table 2.
F1 is the first MFCC with a a filterbank size of 32. It is calculated as described in [18] with the preemphasis coefficient $\alpha = 0.95$ and the Liftering parameter $L = 22$. The calculation of F2 is performed by the Matlab command `lpc(currentFrame,p)` and `dsp.LPCToCepstral`, using a filter order of $p = 12$. F3 is motivated by "mel-cepstability" in [19] and therefore called Modified Mel-Cepstability (MMC) (eq. 5). The main difference to [19] is the dynamic, as the feature is calculated for every event, which is detected with the method described in 2.2. When classifying an event, the average energy of the last 10 seconds is used to normalize the variance of the 12 MFCCs. Thus, the past energy values E_s are stored in a circular buffer, with the corresponding size M_1. The MFCCs are selected from the frame with the highest mean energy value of the currently analysed event ($\mathbf{MFCCs}_{maxEn}$).

$$MMC = \frac{var(\mathbf{MFCCs}_{maxEn})}{\frac{1}{M_1}\sum_{s=1}^{M_1} E_s} \qquad 0 < m \leq M_1 \qquad (5)$$

F4 (CLS) is defined by equation 6 and is based on the Zwicker Loudness as explained in [20], where $N'(z)$ is the specific loudness within a Bark group z. The audio signal is scaled to a level of 30 dB [21], which refers to a whispering sound. A time window of 300 ms for this feature is necessary, to not violate the filters' setting time.

$$CSL = \frac{N'(4) - N'(3)}{max(N'(z))} \quad 0 \leq z \leq 24 \text{ [Bark]} \qquad (6)$$

		Actual value				
		0	**1**	**2**	**3**	**4**
Set to 0		195036 Frames 50.86 %	1551 Frames 0.4 %	1220 Frames 0.32 %	2423 Frames 0.63 %	6212 Frames 1.62 %

Fig. 3. Confusion matrix of the noise detection during preprocessing

The amplitude ratio feature F5, is used as described in [22] (eq. 7) where P_m and N_m refer to the absolute value of the biggest and smallest amplitude of the frame, respectively. However, in this work M_2 is defined by the size of a circular buffer, which stores the frames of the audio signal of the last $20\,\mathrm{s}$.

$$PNAR = Var\left(\frac{P_m - N_m}{P_m + N_m}\right) \quad m = 1, ..., M_2 \qquad (7)$$

2.5. Classification

The classification system consists of 5 binary RBs using the one-vs.-rest strategy and an artificial neural network for voting on the outputs.

To tackle the problem of class noise [23], 5 RBs [2] are implemented in parallel. Each classifier has a tree size of 300 and the following error goals (EG): $EG_{SI} = 7\,\%$, $EG_{SE} = 41\,\%$, $EG_{BI} = 23\,\%$, $EG_{BE} = 23\,\%$, $EG_{NS} = 28\,\%$. EG are determined using heuristic models with the features of table 2. As the test data is highly imbalanced (tab. 1), each RB is trained with a balanced dataset by downsampling the majority class. For voting on the output, an ANN with one hidden layer and 5 neurons is used. Training of the ANN is performed with balanced datasets and scaled conjugate gradient backpropagation (`trainscg`). Size and training method are determined by choosing the best result (lowest classification error), of all possible combinations of 3 to 30 neurons and the most promising pattern recognition training methods `trainscg`, `traincgb`, `trainlm` and `trainrp`.

3. RESULTS

The results of the noise detection (see 2.2), are shown in table 1 and in the confusion matrix (figure 3). All in all $53.83\,\%$ of the data is discarded, with a loss in information of $2.98\,\%$. Over $50\,\%$ of this loss is caused by BE.

Out of 899 features, 5 are selected using the implemented feature selection methods.

The classification results are generated performing a 10-fold crossvalidation on the preprocessed dataset (tab. 1). Classification results are shown in table 3 and in the confusion matrix (tab. 4), respectively.

Table 3. Classification results

	SI	**SE**	**BI**	**BE**	**NS**
Se	80.91 %	80.01 %	34.12 %	66.45 %	29.57 %
Sp	83.56 %	91.70 %	90.53 %	83.32 %	93.52 %
PPV	68.11 %	47.50 %	11.55 %	31.24 %	80.53 %
Error	17.23 %	9.31 %	11.27 %	18.41 %	36.94 %

Table 4. Confusion matrix of the classification result

		Predicted outcome				
		SI	**SE**	**BI**	**BE**	**NS**
Actual value	**SI**	80.92 %	2.96 %	4.54 %	5.44 %	6.13 %
	SE	4.25 %	80.01 %	7.49 %	4.07 %	4.19 %
	BI	13.42 %	34.61 %	36.12 %	9.71 %	6.14 %
	BE	13.05 %	4.05 %	6.89 %	66.45 %	9.56 %
	NS	19.58 %	10.78 %	13.52 %	26.58 %	29.54 %

4. DISCUSSION AND CONCLUSION

Audio files of each smartphone differ in sound quality, snoring characteristics and distance from the source. Classifying these highly imbalanced, class noise data sets into 5 classes is a challenging task. The results in table 4 reflect the difficulties, when labelling the data. BI was hardly be heard, which was made even more difficult after perceptual adaption by a precedent and loud SE sound. Using physiological parameters (e.g. diaphragmatic breathing) for training of the classifiers, should improve the results especially for BI and NS. Furthermore, a more sophisticated method for balancing the classes should be applied.

The loss in information caused by BE, can be explained by the adaptive energy threshold of the noise reduction, eliminating $21.3\,\%$ of this event after a loud SI sound.

All of the selected features are derived from the psychoacoustic domain. This is reasonable, since labelling is based on human perception only. The selected features are legitimized by current research on breath sound detection [1], [19], [22].

In this paper it is shown, that it is possible to classify overnight smartphone audio files in 4 breath classes and 1 noise class. The data is collected with an observational study from 7 different private surroundings, using their own smartphones. Five RBs tackle the problem of class noise and an ANN is applied for voting on the RB outputs. The 5 features which are used for every RB as an input vector, have been automatically selected from a subset of 899 features. To reduce computational time and improve accuracy, adaptive noise suppression and filtering is applied.

Since computational power of smartphones is rising, implementation of the proposed improvements in combination with using more data for training and testing, sleep-disordered breathing detection for in-home becomes more accurate.

5. REFERENCES

[1] Eliran Dafna, Ariel Tarasiuk, and Yaniv Zigel, "Automatic detection of whole night snoring events using non-contact microphone," *PLoS ONE*, vol. 8, no. 12, pp. 1–13, 2013.

[2] Yoav Freund, "A more robust boosting algorithm," May 2009.

[3] Iber C, S Redline, and Kaplan Gilpin A, "Polysomnography performed in the unattended home versus the attended laboratory setting–Sleep Heart Health Study methodology.," *Sleep*, vol. 27, no. 3, pp. 536–540, 2004.

[4] Telefonaktiebolaget L. M. Ericsson, "Ericsson mobility report," Tech. Rep., June 2013.

[5] Andreas Stippig, Ulrich Hübers, and Markus Emerich, "Apps in sleep medicine," *Sleep and Breathing*, pp. 411–417, 2014.

[6] Krau Mousa, Feldes, "Eingebettetes system zur schnarch-erkennung und schnarch-unterbindung," 2013.

[7] Feldes Kraus, "Schnarcherkennung mit diskreten hidden-markov-modellen," Deutsche Gesellschaft fr Akustik e.V. (DEGA), 2014, Fortschritte der Akustik DAGA 2014.

[8] Ali Azarbarzin and Zahra Moussavi, "Unsupervised classification of respiratory sound signal into snore/no-snore classes," *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, pp. 3666–3669, 2010.

[9] David Flanagan, Mahnaz Arvaneh, and Alberto Zaffaroni, "Audio Signal Analysis in Combination with Non-contact Bio-motion Data to Successfully Monitor Snoring," pp. 3763–3766, 2014.

[10] Hangsik Shin and Jaegeol Cho, "Unconstrained snoring detection using a smartphone during ordinary sleep," *BioMedical Engineering OnLine*, vol. 13, no. 1, pp. 116, 2014.

[11] W D Duckitt, S K Tuomi, and T R Niesler, "Automatic detection, segmentation and assessment of snoring from ambient acoustic data.," *Physiological measurement*, vol. 27, no. 10, pp. 1047–1056, 2006.

[12] Haydar Ankshan and Derya Ylmaz, "Comparison of SVM and ANFIS for snore related sounds classification by using the largest lyapunov exponent and entropy," *Computational and Mathematical Methods in Medicine*, vol. 2013, 2013.

[13] Rajkumar Palaniappan, Kenneth Sundaraj, and Sebastian Sundaraj, "Artificial intelligence techniques used in respiratory sound analysis–a systematic review.," *Biomedizinische Technik. Biomedical engineering*, vol. 59, no. 1, pp. 7–18, 2014.

[14] Kuldip K. Paliwal, J.G. Lyons, and K.K. Wo?jcicki, "Preference for 20-40 ms window duration in speech analysis," in *Signal Processing and Communication Systems (ICSPCS), 2010 4th International Conference on*, Dec 2010, pp. 1–4.

[15] P Scalart and J V Filho, "Speech enhancement based on a priori signal to noise estimation," *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2, pp. 629–632 vol. 2, 1996.

[16] Marko Robnik-Šikonja and Igor Kononenko, ," *Machine Learning*, vol. 53, no. 1/2, pp. 23–69, 2003.

[17] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, p. 46ff, The Springer International Series in Engineering and Computer Science. Springer US, 2012.

[18] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book, version 3.4*, chapter 5, Cambridge University Engineering Department, Cambridge, UK, 2006.

[19] Nir Ben-Israel, Ariel Tarasiuk, and Yaniv Zigel, "Obstructive Apnea Hypopnea Index Estimation by Analysis of Nocturnal Snoring Signals in Adults," *Sleep*, vol. 35, no. 9, pp. 1299–1305, 2012.

[20] Michael Möser, Ed., *Messtechnik der Akustik*, p. 610ff, Springer Berlin Heidelberg, 2010.

[21] J Timoney, T Lysaght, and Marc Schoenwiesner, "Implementing loudness models in matlab," *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFX-04)*, , no. 1, pp. 5–9, 2004.

[22] Takahiro Emoto, Masato Kashihara, Udantha R Abeyratne, Ikuji Kawata, Osamu Jinnouchi, Masatake Akutagawa, Shinsuke Konaka, and Yohsuke Kinouchi, "Signal shape feature for automatic snore and breathing sounds classification," *Physiological Measurement*, vol. 35, no. 12, pp. 2489–2499, 2014.

[23] Sunsern Cheamanunkul, Evan Ettinger, and Yoav Freund, "Non-convex boosting overcomes random label noise," *CoRR*, vol. abs/1409.2905, 2014.