INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

# An ‘End-to-Evolution’ Hybrid Approach for Snore Sound Classification

Michael Freitag[1], Shahin Amiriparian[1,2], Nicholas Cummins[1], Maurice Gerczuk[1], Björn Schuller[1,3]

[1]Chair of Complex & Intelligent Systems, Universität Passau, Germany
[2]Machine Intelligence & Signal Processing Group, Technische Universität München, Germany
[3]Machine Learning Group, Imperial College London, U.K.

freitagm@fim.uni-passau.de, shahin.amiriparian@tum.de

## Abstract

Whilst snoring itself is usually not harmful to a person’s health, it can be an indication of Obstructive Sleep Apnoea (OSA), a serious sleep-related disorder. As a result, studies into using snoring as acoustic based marker of OSA are gaining in popularity. Motivated by this, the INTERSPEECH 2017 ComParE Snoring sub-challenge requires classification from which areas in the upper airways different snoring sounds originate. This paper explores a hybrid approach combining evolutionary feature selection based on competitive swarm optimisation and deep convolutional neural networks (CNN). Feature selection is applied to novel deep spectrum features extracted directly from spectrograms using pre-trained image classification CNN. Key results presented demonstrate that our hybrid approach can substantially increase the performance of a linear support vector machine on a set of low-level features extracted from the Snoring sub-challenge data. Even without subset selection, the deep spectrum features are sufficient to outperform the challenge baseline, and competitive swarm optimisation further improves system performance. In comparison to the challenge baseline, unweighted average recall is increased from 40.6 % to 57.6 % on the development partition, and from 58.5 % to 66.5 % on the test partition, using 2 246 of the 4 096 deep spectrum features.

**Index Terms**: competitive swarm optimisation, evolutionary feature selection, convolutional neural network, snoring, computational paralinguistics

## 1. Introduction

Induced by a blockage of the upper airways during sleep, *Obstructive Sleep Apnea* (OSA) can have numerous detrimental effects on someone’s health, including, for example, periodic hypoxia, or an increased risk for cardiovascular diseases [1]. Snoring is a prevalent symptom for OSA [2], and knowledge of the exact site of vibration and obstruction is key to a targeted medical treatment.

Addressing this task, the INTERSPEECH 2017 ComParE Snoring sub-challenge requires classification of snoring audio samples based on the source of vibration in the upper airways [3]. As for past challenges, a standard feature set is provided, containing 6 373 acoustic features. Whilst it has been shown that the inclusion of more features can boost system performance in computational paralinguistics [4], large feature sets can quickly lead to issues with training time and the ‘curse of dimensionality’.

Therefore, many current approaches reduce the effective dimensionality of the feature space, e. g., by quantising features [5, 6], or with dimensionality reduction techniques such as PCA [7]. Filter-based feature selection approaches are also commonly used in computational paralinguistics [8, 9]. Such approaches typically employ statistical tests or measures, such as information gain [10], the Kolmogorov-Smirnov test [11], canonical correlation analysis [12], or the chi-squared test [13]. However, to date, there has been little work on wrapper-based feature selection methods.

Many heuristic wrapper algorithms, such as stepwise forward selection, struggle on large real-world feature sets, due to the exponential size of the feature set search space [14, Ch. 17]. Evolutionary algorithms are a group of heuristic algorithms, which are better suited to large-scale feature selection [15]. They mimic biological evolution, in that they optimise a set of candidate feature vectors over several iterations, or generations, taking into account the quality of past candidates. Evolutionary feature selection has been used successfully in the audiovisual domain, for instance for instrument recognition [16], voice command recognition [17], classification of infant cries [18], speaker identification [19], and emotion prediction [20, 21].

In this paper, we apply a hybrid approach combining deep image convolutional neural networks (CNNs) and evolutionary feature selection to the INTERSPEECH 2017 ComParE Snoring sub-challenge; noting that as some authors are part of the challenge organisers, we do not officially participate in the challenge. The feature selection algorithm is based on competitive swarm optimisation [22, 23], which promises effective feature selection even on large, high-dimensional data sets. To the best of our knowledge, this is the first time a hybrid approach combining deep image CNNs and swarm optimisation has been applied in computational paralinguistics. It is applied to a novel set of 4 096 low-level features that we extracted in an end-to-end manner from the spectrograms of the Snoring challenge data using a pre-trained image classification CNN. We have shown that, even without subset selection, these features, herein referred to as *deep spectrum* features, are sufficient to outperform the challenge baseline [24].

## 2. Evolutionary Feature Selection

Our feature selection approach is based on *competitive swarm optimisation*, which was proposed as a modification of particle swarm optimisation [25] for large scale optimisation [22].

### 2.1. Background

*Canonical particle swarm optimisation* (PSO) is an optimisation approach inspired by the social swarm behaviour of animals, such as bird flocking, or fish schooling [25]. In PSO, a set of candidate solutions, called particles, is allowed to move through the search space over several time steps, or generations. The particles each maintain a *fitness* value, which measures the quality of the solution they represent, and a *velocity* vector. Once per generation, particles are accelerated towards the solutions with the currently highest fitness values, by changing their velocities accordingly. Subsequently, they are moved a small distance along their velocity vector. After sufficiently many generations, the

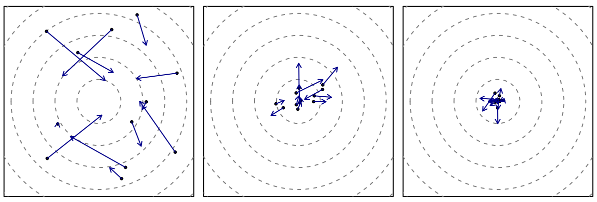

Figure 1: *A particle swarm minimising $f(x) = \exp(\|x\|^2)$ for $x \in \mathbb{R}^2$. Black dots correspond to individual particles, and blue arrows indicate their velocity. As more generations pass (left to right), the particle swarm converges to the optimal solution $x^* = (0, 0)$ of this optimisation problem.*

particle swarm converges at an optimal solution (see Figure 1).

However, PSO has been found to struggle on optimisation problems with many local optima, or high dimensionality [22, 26]. For such large-scale problems, Cheng and Jin have proposed *competitive swarm optimisation* (CSO) [22].

### 2.2. Competitive Swarm Optimisation

Applying the CSO algorithm to wrapper-based feature selection has recently been proposed by Gu et al. [23]. We follow this approach for several reasons. First, as a typical wrapper algorithm, it considers feedback from the machine learning system to fine-tune the feature subset. Therefore, CSO can possibly achieve higher classification performance than a filter-based approach. Second, typical feature sets in computational paralinguistics, as well as our novel deep spectrum feature set, contain thousands of features [8, 24, 27]. The CSO algorithm has been developed specifically for such large-scale optimisation problems [22]. Third, the algorithm naturally lends itself to parallelisation, further alleviating the impact of the feature set size on training time.

Like canonical PSO [25], CSO for feature selection maintains a swarm of particles, representing candidate solutions, which is updated iteratively over several generations. Given a feature set of size $n$, a particle $p = (x, v)$ consists of its location $x \in [0; 1]^n$, and its velocity $v \in [0; 1]^n$. The location $x$ of a particle indicates which features are selected, where feature $i$ is selected iff $x_i > \lambda$ for a constant $\lambda \in [0; 1]$. The velocity of a particle points towards a potentially better solution [23].

Algorithm 1 shows our version of this algorithm. Initially, a predetermined number of particles $n_P$ is randomly initialised (lines 1–3). Subsequently, the swarm is allowed to evolve for several generations. In a given generation $t$, the swarm $P^t$ is first evaluated using the machine learning system (lines 7–13). Given a particle $p = (x, v) \in P^t$, its corresponding feature vector

$$S = \{i \in \{1, \ldots, n\} \mid x_i > \lambda\} \tag{1}$$

is passed to the machine learning system for evaluation. The fitness of the particle $f(p)$ is determined by the classification performance of the machine learning system when using the feature vector $S$. Since the computational cost of CSO is dominated by particle evaluation, a number of optimisations have been applied to this step. Following Gu et al. [23], fitness values are cached in a map $\mathcal{H}$, in order to avoid duplicate evaluations of the same feature vector, which approximately halves the number of required evaluations. The remaining evaluations are performed in parallel, which allows the algorithm to scale well on modern multi-core systems.

**Algorithm 1** The competitive swarm optimisation algorithm

```
 1: for all particles $p_i^0 = \{x_i^0, v_i^0\}$ in swarm $P^0$ do
 2:     initialise position $x_i^0 \in [0;1]^n$ and velocity $v_i^0 \in [0;1]^n$
 3: end for
 4: $n_G \leftarrow$ maximum number of generations
 5: $t \leftarrow 0, \mathcal{H} \leftarrow \emptyset$
 6: while $t < n_G$ do
 7:     for all particles $p_i^t$ in swarm $P^t$ do parallel
 8:         compute feature vector $S_i^t$ from $x_i^t$
 9:         if $(S_i^t, *) \notin \mathcal{H}$ then
10:             evaluate fitness $f(p_i^t)$ using ML system
11:             $\mathcal{H} \leftarrow \mathcal{H} \cup (S_i^t, f(p_i^t))$
12:         end if
13:     end for
14:     while swarm $P^t \neq \emptyset$ do
15:         randomly choose $p_i^t, p_j^t \in P^t$ with $i \neq j$
16:         if $f(p_i^t) > f(p_j^t)$ then
17:             $p_w \leftarrow p_i^t, p_l \leftarrow p_j^t$
18:         else
19:             $p_w \leftarrow p_j^t, p_l \leftarrow p_i^t$
20:         end if
21:         $v_l^{t+1} \leftarrow R_1^t v_l^t + R_2^t(x_w^t - x_l^t) + \phi R_3^t(\bar{x}^t - x_l^t)$
22:         $x_l^{t+1} \leftarrow x_l^t + v_l^{t+1}$
23:         $P^{t+1} \leftarrow P^{t+1} \cup \{p_w^t, (x_l^{t+1}, v_l^{t+1})\}$
24:         $P^t \leftarrow P^t \setminus \{p_i^t, p_j^t\}$
25:     end while
26:     $t \leftarrow t + 1$
27: end while
```

After evaluation, the particle swarm update is performed (lines 14–25), which involves the main deviation from canonical PSO. Instead of learning from the best solution found so far, particles learn from randomly selected competitors. For this, pairwise competitions between particles from two random partitions of the swarm are carried out. Given such a pair of particles, the winner $p_w = (x_w, v_w)$ with the higher fitness score $f(p_w)$ is passed directly to the next generation. The loser particle $p_l = (x_l, v_l)$ updates its location and velocity by learning from the winner particle. In particular, its new location $x_l'$ and velocity $v_l'$ are computed as follows

$$v_l' = R_1 v_l + R_2(x_w - x_l) + \phi R_3(\bar{x} - x_l), \tag{2}$$

$$x_l' = x_l + v_l'. \tag{3}$$

Here, $R_1$, $R_2$, and $R_3$ are vectors in $[0; 1]^n$, which are randomly generated for each generation, $\bar{x}$ is the mean particle location in the current generation, and $\phi \in [0; 1]$ is a constant controlling the influence of $\bar{x}$.

## 3. Experiments and Results

In order to evaluate our approach, we have developed a feature extraction and selection pipeline containing two main components (cf. Figure 2). Feature vectors are extracted from spectrograms using CNNs (Figure 2a), and passed to the CSO algorithm (Figure 2b), which in turn interacts with the machine learning system to determine an optimised feature set.

### 3.1. Database

The Interspeech Snoring sub-challenge is based on the *Munich-Passau Snore Sound Corpus*. This corpus consists of 828 snore samples from four classes relating to the point of obstruction:

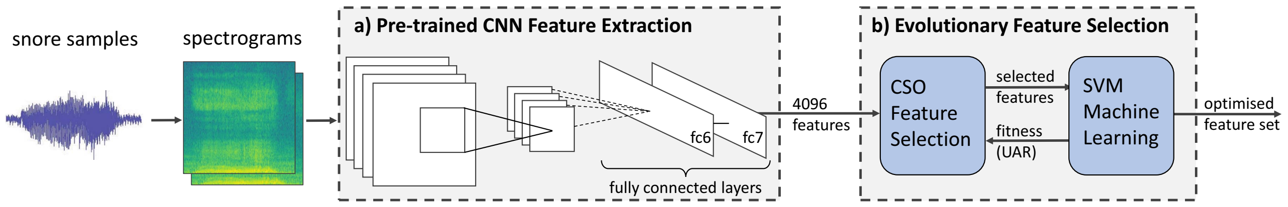

Figure 2: *Block diagram of our system architecture. Spectrograms are generated from whole audio files and plotted with the* matplotlib *Python library. We then use these plots as input for pre-trained CNNs for image processing and extract the activations of fully connected layers as high dimensional feature vectors. Next, we send the extracted feature vectors to the feature selection component which interacts with the machine learning component during optimisation, in order to determine an optimised feature vector.*

*Velum* (V), *Oropharyngeal* (O), *Tongue* (T) and *Epiglottis* (E). For the challenge, the data has been divided equally into train, development and test sets. For a detailed description of the corpus and the class distributions the reader is referred to [3].

It is worth noting that the classes have a very unbalanced distribution; there are considerably more samples labelled *V* than others. Thus, we use the same upsampling strategy as the challenge baseline system [3]. Samples from the *O*, *T*, and *E* classes are replicated proportional to their relative frequency, so that all class labels have roughly the same number of samples.

### 3.2. Deep Spectrum Feature Extraction

Our deep spectrum features are extracted from the spectrogram of each data instance using pre-trained image classification CNNs. First, we extract power spectrograms of every snore sample with Hanning windows of width 16 ms, and overlap 8 ms. Subsequently, they are plotted as images with the Python library matplotlib [28], using a predetermined colour map. The feature vectors are then computed by forwarding these images through pre-trained large deep convolutional neural networks and extracting the activations of a specific fully connected layer. Specifically, we use VGG19 [29] and AlexNet [30] with weights obtained from the Caffe [31] model-zoo[1].

Preliminary experiments indicated that – among the configurations tested (activations of *fc6* and *fc7* of both VGG19 and AlexNet respectively) – the seventh layer (also commonly called *fc7*) of AlexNet using a *viridis* colour map provides the best performance for the Snoring sub-challenge. The resulting feature set therefore has 4 096 attributes (one for every neuron in the fully connected layer) and can be viewed as a high-level representation of spectrograms as seen by AlexNet.

### 3.3. Machine Learning Component

The machine learning system encapsulates a linear SVM. For a given feature vector, the respective features are selected from the training data, and a model is trained on this data. Afterwards, the model is evaluated on the development data, using unweighted average recall (UAR) as the performance measure. Following the challenge rules, we use this performance measure, since the snoring corpus has highly imbalanced class labels [3]. The UAR value is then reported back to CSO as the fitness of the supplied feature vector.

### 3.4. Experimental Settings

The competitive swarm optimiser has been implemented in Java 1.8, using the WEKA machine learning library (version 3.8.1) [32]. The SVM classifiers are trained using the LibLINEAR library with the L2-regularised L2-loss dual solver [33]. We do not normalise or standardise our data, since we found during our preliminary evaluation that this negatively impacts classifier performance. Also, during our preliminary evaluation, the parameters $\phi = 0.1$ and $\lambda = 0.5$ were selected.

Table 1: *System performance on all 4 096 deep spectrum features, i. e., before feature selection, for different SVM complexity values $C$. Performance is measured in unweighted (UAR), and weighted (WAR) average recall on the development partition.*

| $C$ | $10^{-6}$ | $10^{-5}$ | $10^{-4}$ | $10^{-3}$ | $10^{-2}$ | $10^{-1}$ |
|---|---|---|---|---|---|---|
| UAR[%] | 29.8 | 36.7 | 44.8 | 47.3 | 42.7 | 40.5 |
| WAR[%] | 31.1 | 42.8 | 55.1 | 60.4 | 62.2 | 62.2 |

Table 1 shows SVM performance on the development partition for different SVM complexity values $C$. Since UAR peaks for $C$ around $10^{-4}$ to $10^{-3}$, we choose values from this range for further evaluation. The optimal choice of $C$ may change during subset selection, and thus, we report SVM performance on the development partition after feature selection with different numbers of generations and swarm sizes (see Table 2). In order to keep computation time under control, and to eliminate one optimisation dimension, we have limited the number of evaluations in CSO, i.e., the product of the number of generations and the swarm size, to 40 000.

### 3.5. Results

First and foremost, we observe in Table 2 that CSO increases the UAR on the development partition substantially across all configurations, achieving as much as 65.3 % UAR. Furthermore, the algorithm reduces the feature subset size to as little as 48.3 % of its original size. In fact, the feature subset size and the UAR on the development partition are strongly negatively correlated (Pearson, $\rho = -0.89$), meaning that smaller subsets generally achieve higher performance.

Moreover, we also observe that there is generally little variation in UAR on the development partition for different values of $n_G$, and $n_P$. According to Cheng and Jin [22], the performance of CSO does not depend on a large swarm size. On the contrary, we even observe slightly lower performance for large swarm sizes ($n_P$ = 300 or 400). However, this may be caused by the low number of generations that we choose in conjunction with these swarm sizes.

On the test partition, the feature sets that achieve the best UAR on the development partition, i.e., $C = 5 \cdot 10^{-3}$ denoted in grey text in Table 2, result in low UAR scores (60.3 % – 60.7 %) on the test partition. We assume that this is due to overfitting

[1]https://github.com/BVLC/caffe/wiki/Model-Zoo

Table 2: *System performance after feature selection for different configurations. As per the challenge rules, performance is measured in terms of unweighted (UAR) average recall. For completeness, we also show weighted (WAR) average recall values, and the size of the selected feature subset relative to the original feature set $s_F$. Evaluations were performed for different SVM complexity values $C$, numbers of generations $n_G$, and swarm sizes $n_P$. The configurations denoted in grey text have been excluded from further investigation due to overfitting to the development partition. Superscripts on the development set scores indicate the indices of the corresponding configurations in Figure 3.*

| | | | | UAR (WAR) [%] | |
|---|---|---|---|---|---|
| $n_G$ | $n_P$ | $C$ | $s_F$ | devel | test |
| 100 | 400 | $1 \cdot 10^{-4}$ | 60.1 | 56.6 (60.1) | – |
| | | $5 \cdot 10^{-4}$ | 76.7 | 53.1 (63.6) | – |
| | | $1 \cdot 10^{-3}$ | 61.8 | 57.1 (68.2) | – |
| | | $5 \cdot 10^{-3}$ | 48.3 | 63.9 (72.1)$^1$ | 60.6 (63.5) |
| 133 | 300 | $1 \cdot 10^{-4}$ | 58.9 | 56.6 (61.5) | – |
| | | $5 \cdot 10^{-4}$ | 79.8 | 52.4 (64.0) | – |
| | | $1 \cdot 10^{-3}$ | 71.2 | 55.5 (67.5) | – |
| | | $5 \cdot 10^{-3}$ | 52.0 | 61.6 (72.1)$^2$ | 60.3 (66.2) |
| 200 | 200 | $1 \cdot 10^{-4}$ | 53.8 | 57.9 (62.5)$^6$ | 64.5 (62.4) |
| | | $5 \cdot 10^{-4}$ | 77.4 | 53.2 (64.0) | – |
| | | $1 \cdot 10^{-3}$ | 66.8 | 57.3 (70.3) | – |
| | | $5 \cdot 10^{-3}$ | 56.3 | 62.1 (73.9)$^3$ | 60.7 (63.9) |
| 300 | 134 | $1 \cdot 10^{-4}$ | 54.8 | 57.6 (62.5)$^7$ | **66.5** (62.4) |
| | | $5 \cdot 10^{-4}$ | 77.1 | 53.2 (64.3) | – |
| | | $1 \cdot 10^{-3}$ | 64.2 | **58.1** (70.7)$^8$ | 65.3 (65.4) |
| | | $5 \cdot 10^{-3}$ | 49.3 | 65.3 (74.9)$^4$ | 60.3 (63.9) |
| 400 | 100 | $1 \cdot 10^{-4}$ | 56.0 | 58.0 (63.3)$^9$ | 64.6 (60.8) |
| | | $5 \cdot 10^{-4}$ | 68.5 | 54.4 (65.7) | – |
| | | $1 \cdot 10^{-3}$ | 65.3 | 56.3 (68.9) | – |
| | | $5 \cdot 10^{-3}$ | 51.8 | 64.5 (74.9)$^5$ | 60.6 (63.5) |

of the feature subset to the development partition, since particle fitness is derived from performance on the development partition (cf. Section 3.3). Figure 3 shows a plot of development partition UAR over the number of selected features. For all feature subsets selected for $C = 5 \cdot 10^{-3}$, marked by red circles in the plot, the UAR deviates clearly from the distribution of the remaining feature subsets. Generally, the UAR increases little for feature subsets smaller than about 65 % of the original size. For the marked feature subsets, however, the UAR suddenly increases sharply. We interpret this as evidence of overfitting, and exclude the feature subsets obtained for $C = 5 \cdot 10^{-3}$ from further examination.

Among the remaining combinations of $n_G$, $n_P$, and $C$, we choose those four with the next best performance on the development partition for evaluation on the test partition. That way, the best development partition result is 58.1 % UAR, which corresponds to 65.3 % UAR on the test partition. The fourth best configuration performs only slightly worse on the development partition with 57.6 % UAR, but better on the test partition with 66.5 % UAR. Overall, all of them result in considerably higher performance than both the challenge baseline (58.5 % UAR), and the deep spectrum features without subset selection (63.3 % UAR, cf. Table 3). The final feature subset contains 2 246 of the original 4 096 CNN-descriptors.

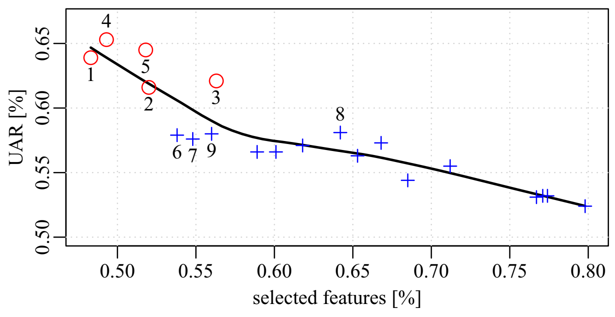

Figure 3: *Relation between selected features ($x$-axis), and unweighted average recall (UAR) on the development partition ($y$-axis). The superscripts in Table 2 map configurations to the corresponding points. Generally, the UAR increases as fewer features are selected, but little improvement is achieved if less than about 65 % are selected. The subsets marked by red circles deviate clearly from this pattern, which is evidence of overfitting.*

Table 3: *Comparison of our proposed approach (Deep Spectrum & CSO) to the challenge baseline (functionals), the end-to-end approach (CNN & LSTM) investigated in the baseline paper, and the purely CNN-based approach (Deep Spectrum). We do not report WAR, since no values are available for the baseline.*

| | | UAR [%] | |
|---|---|---|---|
| Model | Ref. | devel | test |
| Baseline CNN & LSTM | [3] | 40.3 | 40.3 |
| Baseline functionals | [3] | 40.6 | 58.5 |
| Deep Spectrum | [24] | 47.3 | 63.3 |
| Deep Spectrum & CSO | Table 2 | **57.6** | **66.5** |

## 4. Conclusion

This paper proposed a hybrid *'end-to-evolution'* paradigm for snore sound classification. Our results demonstrate that using a large feature set 'as-is' does not necessarily result in optimal system performance. Even though our deep spectrum features are already sufficient to substantially outperform the challenge baseline, feature selection is able to further boost performance on both the development and test partitions.

Naturally, our proposed approach comes with certain drawbacks. Most significantly, like any wrapper algorithm, it increases training time greatly, due to the large number of required particle evaluations. Partly, however, we mitigate this issue by parallelising the evaluation step in CSO. Overfitting, as observed in this paper, is a common problem for wrappers, but we are able to identify and eliminate instances in which overfitting occurs based on their development partition scores.

Based on the results presented, future work will focus on feature selection for computational paralinguistics, such as applying it to regression problems, or to multimodal data. Also, we will investigate the performance of our hybrid approach when fusing features extracted by CNN-descriptors with conventional acoustic feature representations.

## 5. Acknowledgements

This work was supported by the EU's 7$^{th}$ Framework and Horizon 2020 Programmes under grant agreements No. 338164 (ERC StG iHEARu). and No. 688835 (RIA DE-ENIGMA)

## 6. References

[1] P. E. Peppard, T. Young, J. H. Barnet, M. Palta, E. W. Hagen, and K. M. Hla, "Increased prevalence of sleep-disordered breathing in adults," *American journal of epidemiology*, vol. 177, no. 9, pp. 1006–1014, 2013.

[2] M. S. Aldrich, *Sleep Medicine*. Oxford University Press, 1999.

[3] B. Schuller, S. Steidl, A. Batliner, E. Bergelson, J. Krajewski, C. Janott, A. Amatuni, M. Casillas, A. Seidl, M. Soderstrom, A. Warlaumont, G. Hidalgo, S. Schnieder, C. Heiser, W. Hohenhorst, M. Herzog, M. Schmitt, K. Qian, Y. Zhang, G. Trigeorgis, P. Tzirakis, and S. Zafeiriou, "The INTERSPEECH 2017 Computational Paralinguistics Challenge: Addressee, Cold & Snoring," in *Proceedings of INTERSPEECH*. Stockholm, SE: ISCA, 2017, 5 pages.

[4] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "Paralinguistics in speech and language – State-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, 2013.

[5] M. Schmitt, F. Ringeval, and B. Schuller, "At the Border of Acoustics and Linguistics: Bag-of-Audio-Words for the Recognition of Emotions in Speech," in *Proceedings of INTERSPEECH*. San Francisco, US: ISCA, 2016, pp. 495–499.

[6] S. Amiriparian, J. Pohjalainen, E. Marchi, S. Pugachevskiy, and B. Schuller, "Is deception emotional? An emotion-driven predictive approach," in *Proceedings of INTERSPEECH*. San Francisco, US: ISCA, 2016, pp. 2011–2015.

[7] J. R. Williamson, E. Godoy, M. Cha, A. Schwarzentruber, P. Khorrami, Y. Gwon, H.-T. Kung, C. Dagli, and T. F. Quatieri, "Detecting Depression using Vocal, Facial and Semantic Communication Cues," in *Proceedings of the 6th International Workshop on Audio/Visual Emotion Challenge*. Amsterdam, NL: ACM, 2016, pp. 11–18.

[8] B. Schuller, S. Steidl, A. Batliner, F. Schiel, J. Krajewski, F. Weninger, and F. Eyben, "Medium-term speaker states – A review on intoxication, sleepiness and the first challenge," *Computer Speech & Language*, vol. 28, no. 2, pp. 346–374, 2014.

[9] H. Gunes and B. Schuller, "Categorical and Dimensional Affect Analysis in Continuous Input: Current Trends and Future Directions," *Image and Vision Computing*, vol. 31, no. 2, pp. 120–136, 2013.

[10] S. Ultes, A. Schmitt, and W. Minker, "Attention, sobriety checkpoint! Can humans determine by means of voice, if someone is drunk... and can automatic classifiers compete?" in *Proceedings of INTERSPEECH*. Florence, IT: ISCA, 2011, pp. 3221–3224.

[11] A. Ivanov and G. Riccardi, "Kolmogorov-Smirnov test for feature selection in emotion recognition from speech," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing*. Kyoto, JP: IEEE, 2012, pp. 5125–5128.

[12] H. Kaya, F. Çilli, and A. A. Salah, "Ensemble CCA for continuous emotion prediction," in *Proceedings of the 4th International Workshop on Audio/Visual Emotion Challenge*. Orlando, US: ACM, 2014, pp. 19–26.

[13] T. Rahman, S. Mariooryad, S. Keshavamurthy, G. Liu, J. H. Hansen, and C. Busso, "Detecting Sleepiness by Fusing Classifiers Trained with Novel Acoustic Features," in *Proceedings of INTERSPEECH*. Florence, IT: ISCA, 2011, pp. 3285–3288.

[14] R. Bekkerman, M. Bilenko, and J. Langford, *Scaling up machine learning: Parallel and distributed approaches*. Cambridge University Press, 2011.

[15] N. Abd-Alsabour, "A Review on Evolutionary Feature Selection," in *2014 European Modelling Symposium*. Pisa, IT: IEEE, 2014, pp. 20–26.

[16] I. Vatolkin, M. Preuß, G. Rudolph, M. Eichhoff, and C. Weihs, "Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures," *Soft Computing*, vol. 16, no. 12, pp. 2027–2047, 2012.

[17] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *IEEE Transactions on Industrial Electronics*, vol. 52, no. 6, pp. 1478–1489, 2005.

[18] A. Zabidi, W. Mansor, Y. K. Lee, I. M. Yassin, and R. Sahak, "Binary Particle Swarm Optimization for selection of features in the recognition of infants cries with asphyxia," in *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, Pulau Pinang, MY, 2011, pp. 272–276.

[19] M. Zamalloa, G. Bordel, L. J. Rodríguez, and M. Peñagarikano, "Feature selection based on genetic algorithms for speaker recognition," in *2006 IEEE Odyssey – The Speaker and Language Recognition Workshop*. San Juan, PR: IEEE, 2006, pp. 1–8.

[20] H. P. Espinosa, C. A. R. García, and L. V. Pineda, "Features selection for primitives estimation on emotional speech," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. Dallas, US: IEEE, 2010, pp. 5138–5141.

[21] H. Muthusamy, K. Polat, and S. Yaacob, "Particle Swarm Optimization Based Feature Enhancement and Feature Selection for Improved Emotion Recognition in Speech and Glottal Signals," *PLOS ONE*, vol. 10, no. 3, pp. 1–20, 2015.

[22] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 2, pp. 191–204, 2015.

[23] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Computing*, 2016. [Online]. Available: http://dx.doi.org/10.1007/s00500-016-2385-6

[24] S. Amiriparian, M. Gerczuk, S. Ottl, N. Cummins, M. Freitag, S. Pugachevskiy, A. Baird, and B. Schuller, "Snore Sound Classification Using Image-based Deep Spectrum Features," in *Proceedings of INTERSPEECH*. Stockholm, SE: ISCA, 2017, 5 pages.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*. Perth, AU: IEEE, 1995, pp. 1942–1948.

[26] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Proceedings of the Fourteenth International Conference on Machine Learning*, vol. 97. Nashville, US: Morgan Kaufmann Publishers, 1997, pp. 412–420.

[27] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language," in *Proceedings of INTERSPEECH*. San Francisco, US: ISCA, 2016, pp. 2001–2005.

[28] J. D. Hunter, "Matplotlib: A 2D graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[29] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computing Research Repository*, 2014.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.

[31] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*. Orlando, US: ACM, 2014, pp. 675–678.

[32] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[33] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.