Ensemble Bag-of-Audio-Words Representation Improves Paralinguistic Classification Accuracy

Gábor Gosztolya and Róbert Busa-Fekete

***Abstract*—A recently introduced, effective feature extraction technique for computational paralinguistics is that of Bag-of-Audio-Words (BoAW), where we cluster the frame-level training vectors, and represent each speech utterance based on the cluster of its frames. Over the past few years, several improvements have been proposed for the original BoAW approach, but none of them has examined the impact of the stochastic nature of the clustering step. In this study we demonstrate experimentally that the random factor present in the BoAW clustering step is indeed propagated into the next classification step, eventually leading to suboptimal classification performance. As a solution, we propose to train an ensemble of classifiers; that is, we repeat the BoAW codebook selection step several times, train separate classifier models for these BoAW representation versions and combine their predictions. Our results, obtained for three different paralinguistic datasets, demonstrate that this ensemble technique makes the whole paralinguistic classification process more robust, and it leads to improvements in the classification performance. We tested this technique on three different paralinguistic datasets, and achieved the highest Unweighted Average Recall score reported so far on the iHEARu-EAT corpus.**

***Index Terms*—Computational paralinguistics, classification, Bag-of-Audio-Words representation, ensemble learning.**

I. Introduction

COMPUTATIONAL paralinguistics, a subfield of speech technology, consists of tasks that involve identifying phenomena present in human speech besides the actual words uttered. Notable tasks include emotion recognition [1]–[5], conflict intensity estimation [6], [7] and various medical applications like detecting Alzheimer's disease and Parkinson's disease [8]–[12].

One important technical aspect of these tasks is that, to perform utterance-level classification, we need fixed-length feature vectors extracted from recordings of varying lengths. Several techniques have been proposed to solve this problem, such as GMM supervectors [13], i-vectors [14], x-vectors [15] and sequence-to-sequence autoencoders [16], [17]. A competitive alternative to these methods is the Bag-of-Audio-Words (BoAW) representation scheme, inspired by Natural Language Processing and image (video) processing. In the BoAW approach we take the frame-level feature vectors (e.g. MFCCs) of the utterances of the training set and cluster them. Then, for the next step, each frame-level feature vector is replaced by its cluster; utterance-level feature vectors are then calculated as the (normalized) histogram of the clusters corresponding to the frame vectors of each utterance [18]. We can directly use these histograms as feature vectors to perform utterance-level classification and evaluation; for example by using a Support-Vector Machine (SVM, [19]). BoAW representations have been used in various audio processing tasks such as emotion recognition [20], [21], snore sound classification [22] and acoustic event detection [23], achieving competitive results in each case.

Unfortunately, this basic version of the BoAW process is quite sensitive both to the number of audio words (i.e. cluster centers) and to the number of training samples. The former can affect classification accuracy, as using too few clusters may not allow us to represent the utterances in sufficient detail, while too many clusters may lead to overfitting if the clusters are too specific for some actual training utterances. Although computational paralinguistic datasets tended to be only dozens of minutes long, recently larger corpora have been introduced (see e.g. [24]–[26]). This means that, to create the BoAW features, first millions or even tens of millions of frames have to be clustered, leading to enormous execution times.

In order to keep the time requirement of the BoAW codebook creation (i.e. clustering) process within manageable limits, many improvements and simplifications have been proposed. Rawat *et al.* found that using simple random sampling of the input frames as codewords leads to similar accuracy scores as those using clustering for codebook creation [27], while it is evidently significantly faster. Schmitt *et al.* applied the cluster center initialization method of k-means++ for codebook construction [28]. In this method, the first codebook vector is chosen randomly, then each additional center is chosen as the input vector that is the farthest away from the already chosen cluster centers [29]. Following these studies which seek to speed up the codebook construction step, nowadays selecting the cluster centers by random sampling is the most common approach in computational paralinguistics [20], [30]–[32], although some studies employ k-means [33] or GMM clustering [23] as well.

Manuscript received May 26, 2020; revised September 30, 2020 and December 3, 2020; accepted December 4, 2020. Date of publication December 14, 2020; date of current version December 30, 2020. This work was supported by the Hungarian Artificial Intelligence National Laboratory, by the National Research, Development and Innovation Office of Hungary via contract NKFIH FK-124413, and by the grant NKFIH-1279-2/2020 of the Hungarian Ministry of Innovation and Technology. Gábor Gosztolya was also funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences and by the Hungarian Ministry of Innovation and Technology New National Excellence Programme ÚNKP-20-5. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. K. A. Lee. *(Corresponding authors: Gábor Gosztolya.)*

Gábor Gosztolya is with the MTA-SZTE Research Group on Artificial Intelligence of the Hungarian Academy of Sciences, University of Szeged, 6720 Szeged, Hungary (e-mail: ggabor@inf.u-szeged.hu).

Róbert Busa-Fekete is with the Google Inc New York, Google Research, New York, NY 10011 USA (e-mail: busarobi@google.com).

Digital Object Identifier 10.1109/TASLP.2020.3044465

We would like to point out, however, that regardless of what algorithm variation we choose to calculate the BoAW representation of the utterances, it will be sensitive to the randomness involved in the process. Simple random sampling is, without a doubt, a highly stochastic procedure; k-means is reported to be sensitive to cluster center initialization [34]; and the k-means++ initialization step sampling strategy may also prove to be sensitive to the randomly chosen first cluster center, especially for smaller codebook sizes. Although this might seem obvious, surprisingly, earlier studies did not consider this stochastic behaviour as a potential source of suboptimal classification performance. In fact, we found no study at all that investigated the effect of randomness present in the BoAW process. This is why in this paper we will focus on the non-deterministic behaviour of the Bag-of-Audio-Words process.

Our study consists of two key parts. Firstly, we will demonstrate experimentally that paralinguistic classification is indeed adversely affected by the random noise introduced by the BoAW representation. To the best of our knowledge, this is the first study where this is even raised as a hypothesis. Secondly, we will also demonstrate that by training an ensemble learning method (by repeating the BoAW codebook construction process several times), we can make the utterance-level classification process more robust, leading to significant improvements in the classification performance on the test set. In order to demonstrate the above points at the general level, we performed our experiments on three databases, differing greatly in their acoustic conditions, in the phenomenon which had to be detected in them, and in the language of the speakers (German, Hungarian and Australian English).

II. Feature Extraction Methods in Paralinguistic Tasks

Next, to put our work into context, we describe some of the related work of feature extraction approaches utilized in paralinguistic tasks.

In the classification (or regression) step of computational paralinguistic tasks, one speech utterance corresponds to one example. To apply standard machine learning techniques for classification (such as SVM, random forest or DNN), we have to provide a fixed-size feature representation for each utterance. Practically speaking, it involves mapping (and also compressing) a variable-length frame-level feature vector sequence into a fixed-dimensional space.

One of the most widely-used utterance-level feature extraction methods in this area is the 'ComParE functionals' approach. It employs utterance-level statistical functions (mean, standard deviation, percentiles, peak statistics etc.) over the frame-level feature vectors to perform length-normalization. Although it tends to contain several correlated and irrelevant attributes, this set was employed in a wide variety of tasks [6], [35], [36] (even if just to provide a baseline).

Of course, even before the field of 'computational paralinguistics' was defined, there were similar tasks within speech technology. For example, both the speaker identification (i.e. recognizing the actual speaker from a pre-defined list of speakers [37]) and the language identification [38], [39] tasks might fit into the general paralinguistic scheme. For these tasks, several techniques were developed, which share the motivation with Bag-of-Audio-Words that they count the frequency of occurrences of some discrete units such as language-dependent or language-independent phones [40], [41] or prosodic information [42].

Another, quite popular family of feature extraction approaches was developed originally for speaker recognition. Perhaps the most well-known one of these methods, the so-called i-vector technique [14], models 'general speech' by a Gaussian Mixture Model (Universal Background Model, UBM), and expresses speaker and session variability in a compressed space for each speech chunk or utterance. Besides achieving state-of-the-art performance in speaker recognition and speaker identification in its time, i-vectors were also employed in other tasks such as language recognition [43], age determination [35] and detecting dementia from speech [12]. With the rise of deep neural networks, DNN-based speaker recognition approaches such as d-vector [44] and x-vector [15] were introduced; later, these were also applied as feature extractors in paralinguistic (or paralinguistic-like) tasks like age estimation [45], [46], emotion recognition [47] and detecting Parkinson's Disease [48].

III. Bag-of-Audio-Words Feature Extraction

Next, we introduce the Bag-of-Audio-Words representation; for an overview of the BoAW workflow, see Fig. 1. For this, let us denote the frame-level feature vector sequence of an utterance by $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m_i})$, $1 \leq i \leq M$, where $x_{i,j}$ are the d-dimensional feature vectors, and m_i is the length of the ith utterance. First, we pool all the frame-level feature vectors of the utterances of the training set; i.e. we define

$$\mathcal{X} = \bigcup_{i=1}^{M} \bigcup_{j=1}^{m_i} \{x_{i,j}\}. \tag{1}$$

As for the next step, we determine the $W = \{w_1, w_2, \ldots, w_N\}$ set of *codewords*, also being d-dimensional vectors, based on $\mathcal{X}$ (step *codebook construction* in Fig. 1), where $|W| = N$ is a hyperparameter of the BoAW method. For this codebook construction step, multiple methods were defined such as clustering by k-means [18], simple random sampling [27] and using the initialization step of the k-means++ algorithm (usually denoted as `random++`) [28]. Notice that, in the latter two cases, $w_i \in \mathcal{X}$ also holds.

Having obtained the W set of codewords, we can now calculate the Bag-of-Audio-Words representation of any $X_i = x_{i,1}, \ldots, x_{i,m_i}$ utterance. To do this, we first select the closest codeword for each $x_{i,j}$ frame-level feature vector as

$$z(i,j) = \arg\min_{l} \|x_{i,j} - w_l\|_2. \tag{2}$$

This step is usually called *vector quantization* (see Fig. 1). Next, we construct a histogram vector of these values (which we also normalizing by utterance length m_i) as

$$H_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,N}), \tag{3}$$

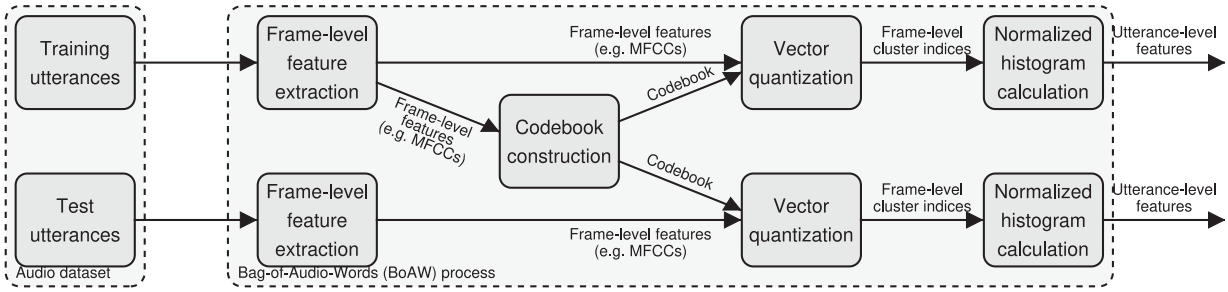

Fig. 1. The schematic general workflow of the Bag-of-Audio-Words feature extraction process.

TABLE I
THE NUMBER OF SPEAKERS AND UTTERANCES IN THE TRAINING AND TEST SETS FOR ALL THREE DATABASES USED

		No. of	No. of	No. of Utterances			Total Duration (h:mm:ss)		
Dataset	Language	classes	speakers (m/f)	Train	Test	Total	Train	Test	Total
Eating Condition	German	7	30 (15 / 15)	945	469	1414	1:52:22	0:59:06	2:51:28
Emotion	Hungarian	4	97 (50 / 47)	831	280	1111	0:20:23	0:07:00	0:27:23
Cognitive Load (*reading sentence* task)	English	3	26 (20 / 6)	1350	600	1950	1:28:21	0:40:58	2:09:19

where

$$h_{i,k} = \frac{\sum_{j=1}^{m_i} \delta\left(z(i,j),k\right)}{m_i} \tag{4}$$

and $\delta(x,y)$ is the Kronecker δ function, which takes the value 1 iff $x = y$ and is 0 otherwise. That is, we calculate the ratio of frame vectors in the current utterance which fall closest to each cluster center. Notice that the size of H_i is independent of the m_i length of the actual utterance, but it is equal to the number of codewords (i.e. N) instead. Also notice that the above H feature vector can be calculated both for utterances used during the codebook construction step and for those not utilized in this process (such as the utterances of the test set).

IV. THE DATABASES USED

We performed our experiments on three different datasets; for the key properties of the corpora, see Table I. The first one was the **iHEARu-EAT** database [49], which contains the utterances of 30 people recorded while speaking during eating. Six types of food were used along with the "no food" class, resulting in seven classes overall. For each speaker and food type, seven utterances were recorded; some subjects refused to eat certain types of foods, resulting in a total of 1414 utterances in German. Although this dataset can be used primarily to test machine learning techniques, Hantke *et al.* anticipated several possible future applications [49]. This dataset was also used in the Interspeech ComParE 2015 Eating Condition Sub-Challenge [50]; we used the official experimental protocol (e.g. training and test set splits). We will refer to this corpus as the **Eating Condition** dataset.

The **Hungarian Emotion Database** [51], used as the second dataset in our experiments, contains sentences from 97 Hungarian speakers who participated in television programmes. A large portion of the segments were selected from spontaneous continuous speech rich in emotions (e.g. talk shows, reality shows), while the rest came from improvised programmes. Note that, although actors tend to overemphasize emotions while acting, in improvisation their performance appears to be more similar to real-life emotions [52].

In this corpus four emotion categories were defined: Anger, Joy, Neutral and Sadness. Unfortunately, previous studies (e.g. [51], [52]) relied on simple ten-fold cross-validation without paying attention to the speaker independence of the folds, as it was not a requirement at the time of recording. To follow the recent trends and to guarantee that the utterances of each speaker are present either during classifier training or evaluation, we defined our custom training and test sets, assigning all utterances of a speaker to either the training or the test set. Our training set consisted of 831 segments, while the test set had 280 utterances. Due to this re-partitioning, our results presented here cannot be directly compared to those presented in the earlier studies (i.e. [51], [52]), but authors reported classification accuracy scores around 66-70%. We will refer to this corpus as the **Emotion** dataset.

The third dataset we used was the **Cognitive Load with Speech and EGG** database [53]; this dataset was created for evaluating algorithms which detect the cognitive load and working memory of speakers during speech. It contains the utterances of native Australian English speakers performing 'span' tasks which require the participants to recall a number of concepts or objects in the presence of distractors. The speakers had to perform three types of tasks. The first one (*reading sentence*) required them to read a series of short sentences, indicate whether each was true or false, and then remember a single letter presented briefly between sentences. Three different cognitive load levels were defined: low when recalling after one sentence, medium when remembering after two sentences, and high after the third, fourth and fifth sentences. The remaining two tasks were variants of the Stroop test [54]: the speakers had to name the font colour of words corresponding to different colour names. In the *Stroop time pressure* task, at the high level the participants had to do this in a very short period of time (0.8s), while in the *Stroop dual task* they had to perform a tone-counting task at the high level besides naming the font colour.

Since the three tasks performed were inherently different, it was advisable to train distinct classifier models for them (for details, see [55]). However, due to the distribution of utterances, this leads to fairly tiny datasets for the two Stroop tasks: from the 1674 utterances of the training set, only 162-162 recordings contain speech recorded during the two Stroop test variations. After considering the tiny size of the two sub-tasks involving the Stroop test, we decided to use only the *reading sentence* task in our experiments.

This dataset was later used in the Interspeech ComParE 2014 Cognitive Load Sub-Challenge [55]; we followed the official evaluation protocol of this dataset with one slight change: we decided to merge the training and development sets (utterances of 6-6 speakers), and we set the hyperparameters in speaker-wise cross-validation. Therefore, our reported test set results are comparable with those found in the literature. We will refer to this database as the **Cognitive Load** dataset.

V. Experimental Setup

Our classification pipeline has a standard structure. First, we extract the frame-level attributes, calculate and standardize the BoAW vectors. Then we choose the hyperparameter vector based on cross-validation performance (over the training set), and evaluate the classifier models on the test set. We also experiment with combining the BoAW-based predictions with those obtained using the 'ComParE functionals' features. Next, we will describe the technical aspects of these steps.

A. BoAW Parameters

We utilized the OpenXBOW package (version 1.0) [56], which is an open-source BoAW toolkit written in Java. It supports various strategies of vector quantization, codebook construction, and normalization of both the input frame-level features and the calculated histograms. We tested codebook sizes of $N = 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192$ and 16384. For codebook construction, we applied the strategy of random sampling from the frames of the training set. Furthermore, following Pancoast and Akbacak, we assigned each frame-level feature vector to the nearest *five* clusters, as it was shown to aid classification performance [57].[1]

We used 12 Mel-Frequency Cepstral Coefficients (MFCCs) along with energy as frame-level inputs; and following preliminary tests, we also included the first-order derivatives. We constructed codebooks for the original and the Δ values independently, and simply concatenated the two BoAW representations, just as suggested by Schuller et al [17], [31]. This approach is reported to improve classification performance, which was reinforced by our preliminary tests. (For consistency, we will always give the combined BoAW codebook size values; i.e. $N = 32$ means 16 clusters for the original 13 MFCC vectors and 16 clusters for the first-order derivatives.) We shall denote this approach by 'BoAW-MFCC' later on.

[1]The command line parameters were `-attributes n1[13]2[13] -a 5 -c random -norm 1 -size N` (`N` being the codebook size).

B. Utterance-Level Feature Preprocessing

Before the classification step, we employed standardization of the utterance-level features (i.e. the BoAW vectors); that is, we applied a linear transformation to convert them so as to have zero mean and unit variance. However, several studies (see e.g. [50], [58], [59]) have demonstrated that, depending on the actual task, speaker-wise feature standardization might assist the subsequent classification step. As our preliminary tests reinforced this finding for the datasets of Eating Condition and Cognitive Load, we standardized all feature sets by applying this approach on these two corpora. We made use of the annotated speaker IDs for the training set, while the speakers of the test set were determined by a standard speaker clustering method. The clustering method of our choice was the Agglomerative Hierarchical Clustering (for the details, see [60]). The number of speakers was determined based on the difference of cluster distances. After speaker clustering, the transformation parameters of the standardization step were calculated for the utterances of each speaker separately; regarding the Emotion dataset, where we employed global standardization, these transformation parameters were determined on the training set and then applied to the test set with the same parameters.

C. The Classification Process

Our classification process followed standard paralinguistic procedures (see e.g. [31]): we applied Support Vector Machines for utterance-level classification, utilizing the LibSVM library [61]. We used the C-SVC method with a linear kernel; the value of C was tested in the range $10^{\{-5,\ldots,1\}}$.

Hyperparameters (BoAW codebook size N and SVM complexity C) were tuned in speaker-wise cross-validation (CV) based on the training set. That is, for a given hyperparameter vector, we withheld the training examples corresponding to the utterances of one speaker during SVM training, and evaluated the trained model on these withheld examples. Repeating this process for all the speakers of the training set, we obtained predictions for all the utterances, which allowed us to evaluate classification performance on the full training set. Next, the hyperparameters (i.e. N and C) leading to the highest-quality classification were used to train a classifier model on the whole training set, which was finally evaluated on the test set. Unfortunately, the Emotion dataset had a fairly large number of speakers, and many of them uttered only a few sentences. Therefore, for this particular dataset we split the training set into 10 (speaker-independent) folds, and set the hyperparameters in 10-fold cross-validation.

We measured the classification performance via the Unweighted Average Recall metric (UAR, [62]), being the mean of the class-wise recall scores. That is, for a confusion matrix $\mathcal{C} = c_{i,j}$ for K classes ($1 \leq i, j \leq K$), where $c_{i,j}$ corresponds to the number of utterances belonging to class i and classified as class j, we define

$$\mathrm{Recall}_i(\mathcal{C}) = \frac{c_{i,i}}{\sum_{j=1}^{K} c_{i,j}}. \tag{5}$$

Following this, we can write

$$\mathrm{UAR}(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K}\mathrm{Recall}_i(\mathcal{C}). \quad (6)$$

The Eating Condition and the Cognitive Load datasets had fairly balanced class distributions, but the Emotion corpus was found to be seriously imbalanced class-wise. We overcame this by employing upsampling for this particular corpus: we repeated the training samples of the rarer classes in the training set to match the size of the most frequent class. We did this for all SVM model training steps, i.e. even during cross-validation.

D. The ComParE Functionals Feature Set

As the standard paralinguistic solution, we utilized the 6373-sized 'ComParE functionals' attribute set (see e.g. [50]). We used the openSMILE tool [63] (version 2.3.0) to extract the features, utilizing the `IS13-ComParE` configuration file. This feature set led to UAR scores of 74.3%, 54.5% and 62.9% in CV, and 74.8%, 60.3% and 63.4% on the test set, Eating Condition, Emotion and Cognitive Load datasets, respectively.

E. Prediction Combination

From our experiences (e.g. [64]) we know that it might be beneficial to use multiple different (utterance-level) feature sets, as these might represent the individual utterances from a different aspect, thus aiding classification. To make use of both the ComParE functionals and the Bag-of-Audio-Words feature sets, we decided to opt for *late fusion* [65]; that is, we trained separate SVM models for the different types of features, and combined the predictions in the second step. Again, following our previous paralinguistic studies, we took the weighted mean of the two posterior estimates; weights were chosen as the ones that led to the best UAR score in cross-validation, determined by a grid search with 0.05 increments.

VI. Experimental Results

Next, we present and analyze our test results on the three paralinguistic datasets.

A. Results

First, we focused on the variance introduced by the Bag-of-Audio-Words representation by examining the classification UAR scores obtained by training our SVM models on BoAW representations extracted with identical hyperparameters. A high variance also has a clear negative impact on classification performance: since the selection of the hyperparameters – the BoAW codebook size N and the SVM complexity C – is done based on classification performance in cross-validation, a high variance of the models means that we are likely to choose a hyperparameter vector which leads to a suboptimal, or even a sub-average classification performance on the test set.

To measure this variance, we created the BoAW codebooks (for each tested codebook size N) repeated for 10 times, each time using a different random seed. We determined the C value for the SVMs by an exhaustive search using BoAW models for three random seed values. Therefore, the trained SVM models differed only in the random seed value used during the BoAW codebook construction process.Fig. 2 shows the average UAR scores we got as a function of codebook size; the error bars indicate the minimal and maximal values. Although it is clear that the scores follow a general trend, it can also be seen that their variance is quite high, even for higher values of N. For the **Eating Condition** dataset, the scores are increasing along with a higher N value, but in cross-validation the difference between the minimal and maximal value is usually between 2.8% and 4.9% (absolute), and for $N = 64$ it is actually 9.9%. (For the test set, these values lie between 1.7% and 6.8%, while for $N = 32$, we got a difference of 9.3% between the best and the worst BoAW-based model.) In the case of the **Emotion** database, the best (average) performance was measured in the range $256 \leq N \leq 1024$, but the individual UAR scores varied to a great extent: even in this interval, the difference between the best and worst measured UAR value is between 6% and 9%, while for lower N values it even reached 11%. For the **Cognitive Load** dataset, the score deviation in general is lower, but the absolute difference is between 2% and 6% in all cases, and for the best region (i.e. $N = 2048$ and $N = 4096$) it is above 4% for the test set, which is actually a large difference for this particular database. These differences support our initial hypothesis that the stochasticity of BoAW codebook construction also leads to a high variance in classification performance, both in cross-validation and on the test set.

When we examine how the UAR values behave after (late) fusion (see Fig. 3 for the average of the fused scores; the error bars again indicate the minimal and maximal UAR values), we might notice that, in contrast with Fig. 2, the scores are generally quite high for most N values. This is obviously so because the combination with the 'ComParE functionals' approach increased the robustness of the (combined) predictions (again, compared to the previous experiment, i.e. Fig. 2). Still, there is a relatively high variance of the UAR scores for each N, although it is somewhat smaller than for the original scores.

Table II summarizes the best UAR scores measured. Besides the 'ComParE functionals' approach, we list four variations of BoAW-MFCC. 'Single' refers to the typical setup when using Bag-of-Audio-Words: using only one BoAW model, calculated by using only one single random seed value (in our experiments simulated by taking the first random seed value tested[2]). 'Average' refers to the mean of the UAR values of the ten BoAW models (extracted using the same N value, but with a different random seed). Finally, 'maximal (CV)' and 'maximal (test)' denote the cases where we choose the classifier model (practically the hyperparameters C, N and the random seed) which give the highest UAR value in cross-validation and on the test set, respectively.

We would like to point out that now we are interested in the variance of the BoAW-based models, and we selected the cases examined on this basis. Therefore, not all the listed cases represent an approach which can be applied in practice; for

[2]It was 117441911

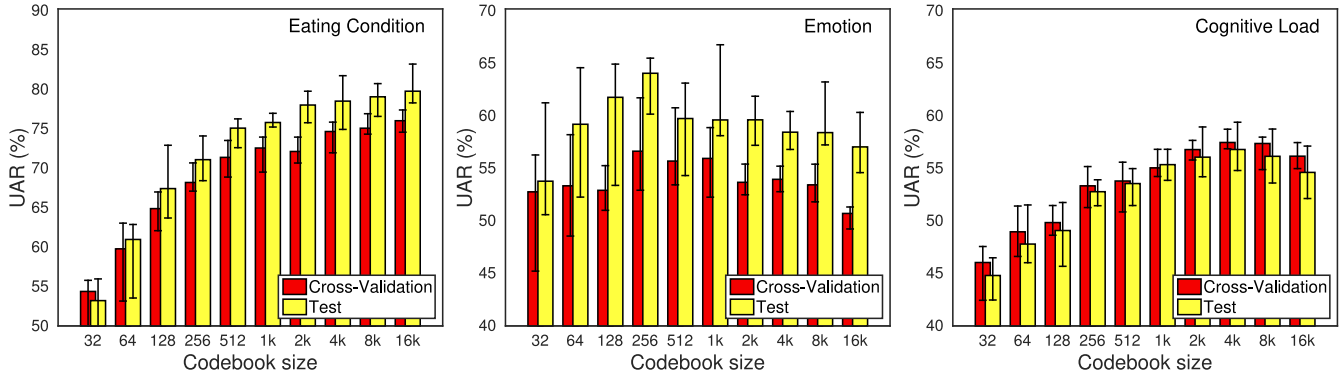

Fig. 2. The average UAR scores obtained using the BoAW-MFCC representation for the different codebook sizes and for all three databases. The error bars indicate the minimum and maximum values.

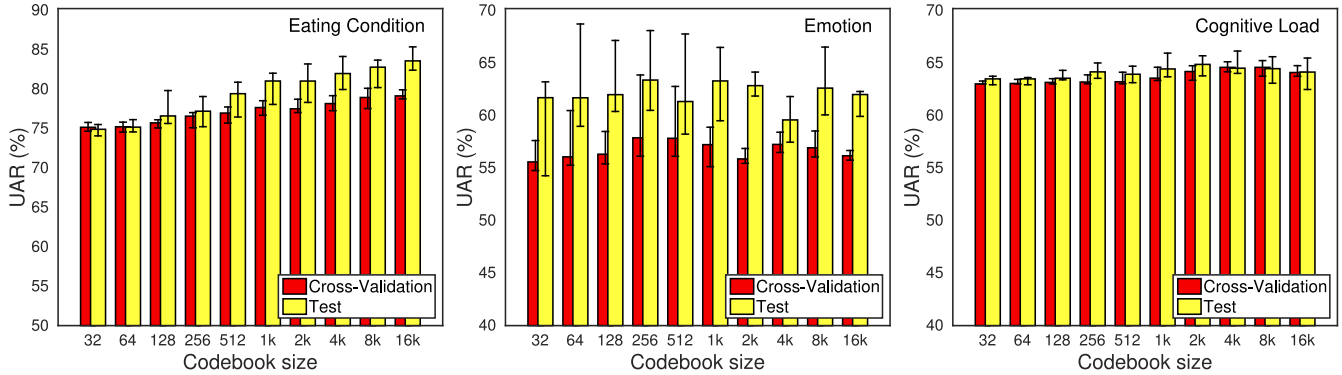

Fig. 3. The average UAR scores obtained by combining the ComParE and the BoAW-MFCC representations for the different codebook sizes and for all three databases. The error bars indicate the minimum and maximum values.

TABLE II
THE UAR SCORES OBTAINED USING THE VARIOUS BOAW-MFCC APPROACHES, WITH AND WITHOUT A COMBINATION WITH THE BASELINE COMPARE FUNCTIONALS APPROACH

	Eating Condition		Emotion		Cognitive Load	
Feature Set	**CV**	**Test**	**CV**	**Test**	**CV**	**Test**
ComParE functionals	74.3%	74.8%	54.5%	60.3%	62.9%	63.4%
BoAW-MFCC, single	77.3%	78.2%	56.2%	52.1%	57.5%	53.6%
BoAW-MFCC, average	76.1%	79.8%	58.0%	61.5%	57.7%	56.1%
BoAW-MFCC, maximal (CV)	77.3%	78.2%	61.6%	65.2%	58.6%	55.3%
BoAW-MFCC, maximal (test)	74.5%	83.1%	57.2%	66.7%	56.8%	59.3%
ComParE + BoAW-MFCC single	79.4%	82.3%	56.9%	62.7%	64.7%	64.4%
ComParE + BoAW-MFCC average	79.3%	83.0%	59.6%	63.4%	64.7%	64.4%
ComParE + BoAW-MFCC maximal (CV)	80.0%	81.4%	63.8%	63.7%	65.1%	65.1%
ComParE + BoAW-MFCC maximal (test)	78.7%	85.2%	55.7%	68.6%	64.4%	66.1%

example, choosing the classifier model which leads to the best performance on the test set ('BoAW-MFCC, maximal (test)') is clearly not something one could do in ordinary classification experiments, but it makes sense to include this specific model when investigating model variance.

Examining the values obtained (see Table II), we can see that the approach 'BoAW-MFCC, single' (i.e. the typical approach when using Bag-of-Audio-Words; practically, our baseline) in fact led to similar UAR scores to those with the BoAW-based models on average (line 'BoAW-MFCC, average'), with a difference of 1.2% (absolute) in the CV setup and 1.6% on the test set for the **Eating Condition** corpus. Choosing the BoAW model which led to the highest CV UAR score (case 'BoAW-MFCC, maximal (CV)'), however, led to a sub-average performance on the test set; while when we chose the best-performing model (in fact, random seed) on the test set (an outstanding value of 83.1%), the CV performance was below average. We regard this finding alone as an indicator of a high level of stochasticity.

Examining the results on the **Emotion** and on the **Cognitive Load** datasets, we can draw similar conclusions. When we combined our predictions with those obtained by using the 'ComParE functionals' feature set, we can find an even smaller

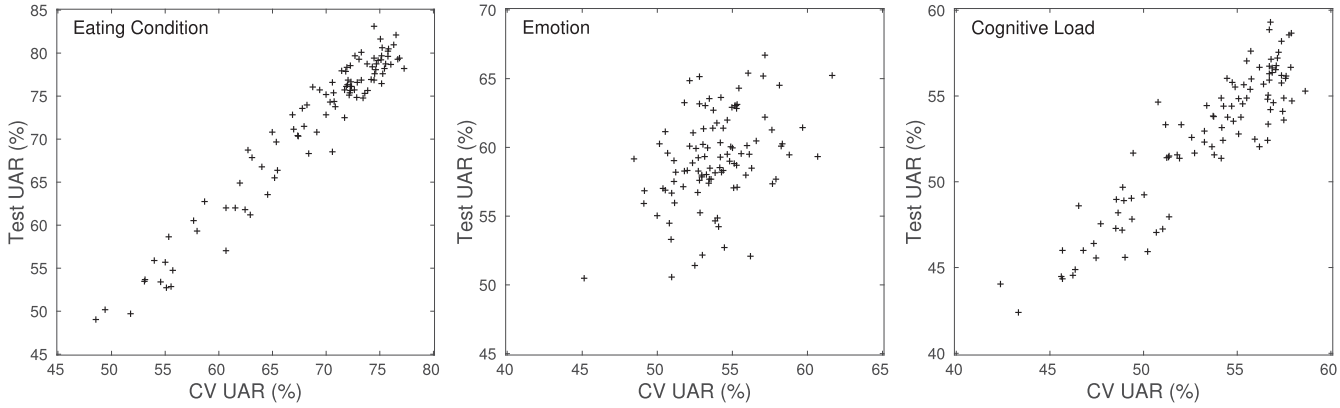

Fig. 4. Absolute UAR scores in cross-validation and on the test set for the different BoAW models tested.

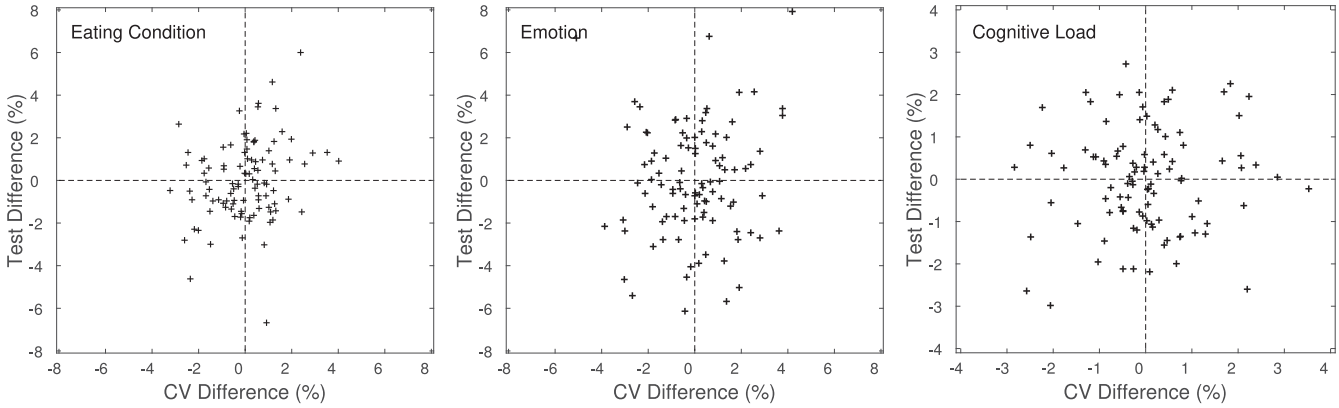

Fig. 5. UAR score differences compared to the average UAR score for the same codebook size in cross-validation and on the test set for the different BoAW models tested.

connection among the UAR values in cross-validation and on the test set: for example, the best-performing model on the test set (line 'maximal (test)') was worse than the average score of the ten models in cross-validation on all three datasets.

In our view, this behaviour is present because the random factor affecting the BoAW codebook construction process affects the final UAR scores in a random way as well. This means that a slight performance increase in the CV setup *due to a different random seed* does not guarantee a similar (or, in fact, any) increase of the UAR score on the test set. This also means that, if one uses only one BoAW model for each N value and selects N as the one leading to the highest UAR value in cross-validation (which is the standard practice), one might end up with a suboptimal codebook size (in terms of test set performance). To verify this hypothesis, we carried out another experiment.

B. Relation of the Cross-Validation and Test Set Predictions

Fig. 4 shows the UAR scores obtained for the 100 models (10 random seed values tested for each of the 10 codebook size variations). At first glance, the values for the Eating Condition and the Cognitive Load datasets seem to be highly correlated, suggesting that a high performance in cross-validation also leads to a well-performing model on the test set. (These values have a Pearson's correlation coefficient of 0.972 and 0.914, Eating Condition and Cognitive Load tasks, respectively, also indicating a high level of correlation.) In contrast, for the Emotion corpus we cannot find such a strong connection (correlation coefficient: 0.418); this might be explained, however, by the subjectivity of annotation leading to label noise, making this task (and emotion detection in general) harder for automatic methods. In contrast, the class labels of the other two datasets were determined objectively.

Notice, however (see Fig. 2), that the number of audio words (i.e. N) has a great influence on the classification performance. To eliminate this effect of codebook size, next we adjusted the UAR scores: for each configuration, we reduced the measured UAR values by the *average UAR score* obtained for the appropriate codebook size. (This way we measured how much better or worse the performance of the given configuration was than the average BoAW configuration with the same codebook size.) The resulting difference values, showing the relative effect of each given random seed value in cross-validation and on the test set, can be seen in Fig. 5. Inspecting this figure, it is quite apparent that there is no more than a slight connection among the relative performance gain on the two database subsets. (We measured Pearson's correlation score of 0.454, 0.188 and 0.167 for these values, for the Eating Condition, Emotion and Cognitive Load tasks, respectively, which also reflect a low level of (linear)

dependence.) In contrast, the *mean* UAR scores of the CV and test UAR scores (measured for the different codebook sizes) had correlation values of 0.996, 0.693 and 0.992.

Repeating this experiment for the combined models, we were able to draw similar conclusions: after removing the influence of the codebook size, we found that the UAR scores on the two subsets are only loosely correlated, as we measured Pearson's correlation scores of 0.368, 0.056 and 0.049, Eating Condition, Emotion and Cognitive Load datasets, respectively. We interpret these results as they also support our hypothesis that the random factor (inherently present in the BoAW codebook construction step) adds some random noise to the performance of the next classification step, which may affect the different examples, hence the different database subsets independently. As model and hyperparameter selection (i.e. BoAW codebook size, combination weights) are typically carried out in cross-validation or on a development set, this noise is likely to lead to a suboptimal performance on the independent test set.

VII. Ensemble BoAW Representation

In the previous section we showed that by using the BoAW procedure we inherently introduce some random noise into the UAR scores, which makes model selection really challenging. Next, we will show that the effect of this high model variance can be reduced by training an ensemble of the models built using different random seeds.

A. Ensemble Learning

The basic principle of ensemble learning is to train several different, but similar machine learning models, and combine their outputs in some way. Perhaps the best-known such techniques are *bagging* (or *bootstrap aggregation*) and *boosting*. Bagging carries out the training of such similar models by randomly selecting *subsets* of the training data [66]. Boosting, in contrast, trains the next individual classifier model by focusing on training instances which were mis-classified by previous models (e.g. by using larger weights for these examples, [67]). *Stacking*, another ensemble learning technique, is basically a two-step learning scheme, where different classifier models (for example, different algorithms) are trained on the whole training data, and their outputs are combined via another machine learning method [68]. Notice that all these ensemble approaches use the same feature representation, and the difference in the classifier models trained are due to using some subsampled data or weighting.

Our approach differs from these ensemble approaches in the sense that the individual learners use different feature sets, which depend on some random initialization (i.e. generating the codewords of BoAW). Due to this, we find the proposed mechanism more related to *random projection* instead. In random projection, we map our feature vectors to a lower-dimensional space in a controlled random manner, e.g. by multiplying them with a Gaussian random matrix [69] or with a sparse random matrix [70]. Since this process is stochastic by nature, if we repeated this procedure multiple times, we would end up with several different *representations* of the *full training data*, eventually leading to a difference in the classifier models trained on them. We find the BoAW representation to be similar to random projection because of the stochasticity inherently present in the feature representation.

B. The Ensemble BoAW Model

Regarding the Bag-of-Audio-Words process in paralinguistic audio classification, we propose calculating the BoAW codebook several times using the same parameters, but each time applying a different random seed. This eventually leads to a number of different representations ("projections") of the same training data. Although in theory concatenating these feature vectors and training only one classifier model might lead to a more robust performance than relying on any of the individual representations, we would end up with unrealistically huge feature vectors, which might prove to be unfeasible in practice. Therefore we chose to train separate classifier (e.g. SVM) models on these BoAW representations in the next step. Of course, as we showed in Section VI, we may expect the performance of these models to have a high variance. To make the predictions more robust (and thus, hyperparameter selection more reliable), we suggest simply averaging out the posterior scores got after classifier evaluation in an unweighted manner. Formally, we calculate the posterior estimate provided by the ensemble model as

$$P_e(c_i|X) = \frac{1}{m}\sum_{j=1}^{m} P_j(c_i|X) = \frac{1}{m}\sum_{j=1}^{m} P_j(c_i|H^j), \qquad (7)$$

where c_i denotes the ith class ($1 \leq i \leq K$), X is the frame-level feature sequence of the actual utterance, H^j is the BoAW representation of X calculated by the jth BoAW model, and the P_j value is the individual posterior estimate provided by the jth SVM model. We call this approach the 'Ensemble BoAW approach'.

In our experiments, the number of models in the ensemble (m) was set to 10. We repeated this procedure for all three datasets and all the tested N values. To set the complexity hyperparameter of SVM, first we performed the proposed procedure with $m = 3$ classifier models, and then we chose the C value that led to the best performance for the given codebook size in cross-validation.

C. Bayesian View

Our approach can naturally be viewed as a Bayesian classifier combination mechanism where the prior distribution μ is determined by the random selection of the codewords $W = \{w_1, \ldots, w_N\}$. More precisely, let us denote the classifier method (including the feature extraction step) by g. The function g represents an end-to-end classification pipeline that maps the input sequence of utterance to a score value. In our actual realization, the g function represents the feature extraction carried out by BoAW and, in addition to this, the classifier itself (which is an SVM). The input parameters of function g, among many, is the dictionary of codewords, therefore we shall write $g(X; W)$. For the sake of presentation, the rest of the input parameters of g, which are often called hyperparameters, are concealed. With

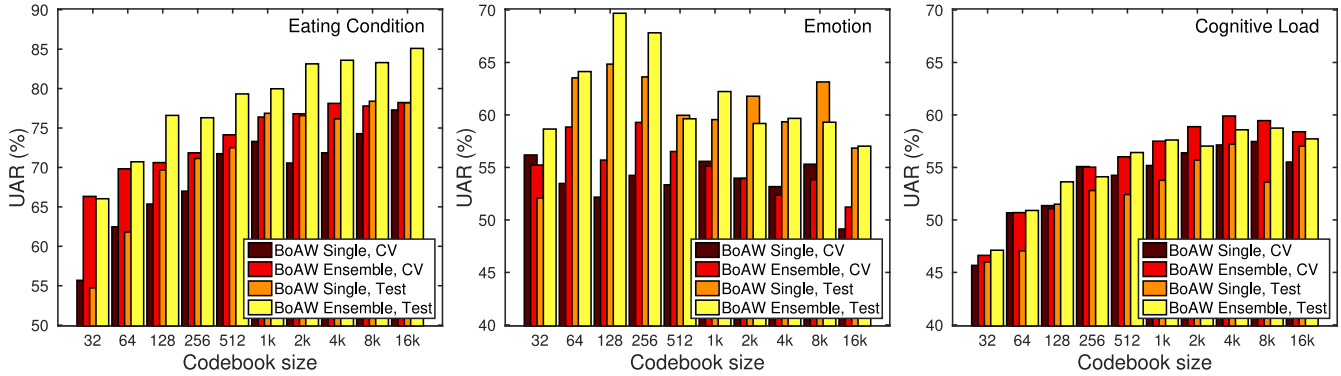

Fig. 6. The UAR scores obtained using the 'single' and the 'ensemble' BoAW-MFCC approaches for the different codebook sizes and for all three databases.

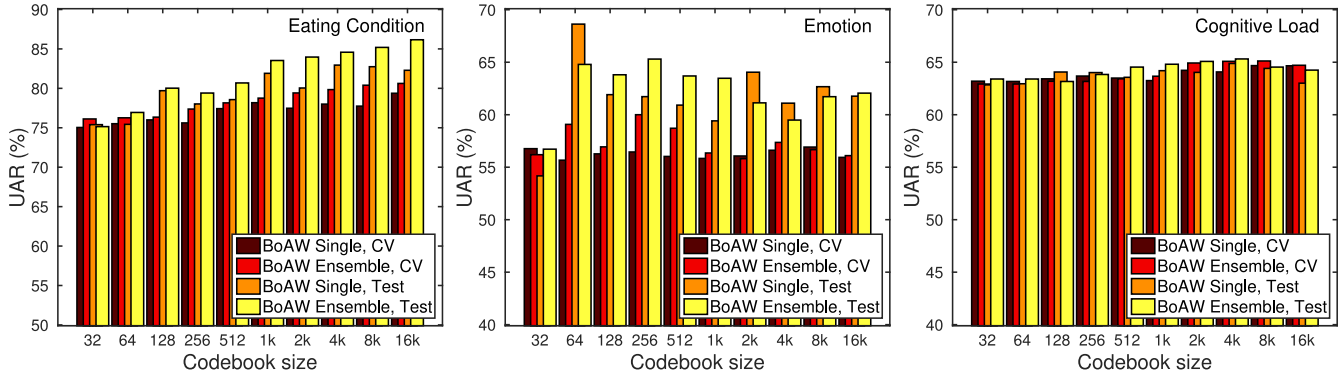

Fig. 7. The UAR scores obtained by combining the ComParE and the 'single' and the 'ensemble' BoAW-MFCC approaches for the different codebook sizes and for all three databases.

this notation in hand, our approach can be viewed as an estimate of the following Bayesian mixture of classification methods in the limit:

$$g(X) = \int g(X;W)\mu(W), \tag{8}$$

where $\mu(W)$ is a distribution of the possible codewords of N elements. We believe that $\mu(W)$ explains the vast part of the variance in the performance of the classifiers using the BoAW approach. Our empirical estimate of (8) which we used in our experiments is

$$\hat{g}(X) = \frac{1}{m}\sum_{i=1}^{m} g(X;W_i), \tag{9}$$

where m is the number of models that are combined.

D. Classification Results

Fig. 6 shows the UAR scores achieved in cross-validation and on the test set using one BoAW model ('BoAW Single') and using the proposed, ensemble BoAW approach ('BoAW Ensemble') for the three datasets used. It is quite apparent that the proposed ensemble classification model outperforms the single BoAW one: the UAR values we achieved are higher in almost every case. When combining the BoAW predictions with the ComParE ones (see Fig. 7), a similar trend is visible.

Examining the configurations which proved to be the best in cross-validation (see Table III), the performance difference is perhaps even more obvious. We can also see that combining the posterior values of the BoAW models calculated by using different random seeds indeed stabilizes the predictions, especially on the Eating Condition and on the Cognitive Load corpora. Overall, we achieved absolute UAR improvements of 6.9%, 15.7% and 5.0%, corresponding to relative error reduction (RER) values of 31%, 33% and 11% on the Eating Condition, Emotion and Cognitive Load corpora, respectively. We also notice that the combined score significantly outperformed the one obtained by using the BoAW model with one random seed (line 'ComParE + BoAW-MFCC single'): treating the latter value as our baseline, we achieved relative error reduction scores of 22%, 12% and 3%, Eating Condition, Emotion and Cognitive Load corpora, respectively. (We find this approach more appropriate for use as a baseline than the average performance of the BoAW models, because this is the standard solution described in the literature.) We would also like to add that our UAR score of 86.2% achieved on the Eating Condition corpus is the highest one published so far, significantly exceeding the UAR value of 83.1% obtained by Kaya *et al.* [58].

TABLE III
THE UAR SCORES OBTAINED USING THE SINGLE AND ENSEMBLE BOAW-MFCC APPROACHES, WITH AND WITHOUT A COMBINATION WITH THE BASELINE COMPARE APPROACH

Feature Set	Eating Condition		Emotion		Cognitive Load	
	CV	Test	CV	Test	CV	Test
ComParE functionals	74.3%	74.8%	54.5%	60.3%	62.9%	63.4%
BoAW-MFCC, single	77.3%	78.2%	56.2%	52.1%	57.5%	53.6%
BoAW-MFCC, average	76.1%	79.8%	58.0%	61.5%	57.7%	56.1%
BoAW-MFCC, ensemble	78.2%	85.1%	59.3%	67.8%	59.9%	58.6%
ComParE + BoAW-MFCC single	79.4%	82.3%	56.9%	62.7%	64.7%	64.4%
ComParE + BoAW-MFCC average	79.3%	83.0%	59.6%	63.4%	64.7%	64.4%
ComParE + BoAW-MFCC ensemble	80.6%	86.2%	60.0%	68.2%	65.1%	65.3%

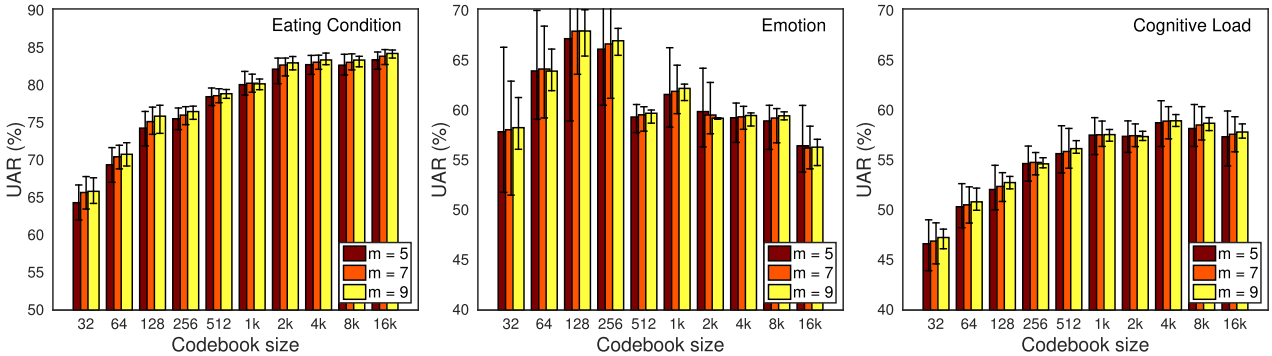

Fig. 8. The average UAR scores obtained using the Ensemble BoAW approach with different number of models (m) combined. The error bars indicate the 5th and 95th percentile values.

E. The Variance of The Ensemble Models

The proposed 'Ensemble BoAW', by combining the predictions of individual BoAW-based classification processes, led to improved scores by making the predictions more robust. Still, since it is a combination of a finite number of classifiers, it might be worth taking a look at the variance of the ensembles. To do this, we took the same 10 SVM models for each N value as before, and constructed ensembles of them with parameters $m = 5$, $m = 7$ and $m = 9$ for every possible combination.

Fig. 8 shows the mean UAR values on the test set of the three corpora as a function of codebook size N; the error bars show the 5th and 95th percentiles of the measured scores. Besides noticing that the shown UAR scores have a similar tendency as the ensemble scores for $m = 10$, we can also see that the value of m has little effect on the mean UAR score. The largest absolute difference was only 1.6% for the Emotion dataset ($N = 128$), while for the Emotion and the Cognitive Load corpora it was less than 1% ($N = 256$ and $N = 128$, respectively).

However, we observed differences regarding model variance. Firstly, increasing N leads to smaller differences among the reported extreme values, even for the Emotion dataset (where such differences tend to be larger than in the case of the other two corpora). Secondly, increasing m leads to a smaller variance; for example, for the Eating corpus, the difference between the 95th and the 5th percentile UAR was between 2.3% and 4.7% for $m = 5$; for $m = 7$ it was $1.9\% \ldots 4.4\%$, while it dropped to $1.1\% \ldots 3.4\%$ in the $m = 9$ case. This, in our view, indicates that by increasing the number of models in the ensembles, we can reduce the level stochasticity, therefore making the predictions more robust.

VIII. SUMMARY AND CONCLUSIONS

The Bag-of-Audio-Words (or BoAW) representation is an audio feature extraction approach, which was previously employed in several computational paralinguistic tasks, and it achieved competitive results. Although in the literature several modifications and improvements have been proposed for the BoAW scheme, none of them has altered its essentially stochastic nature. Surprisingly, we found no study at all in the literature that addresses the influence of the random factor of the BoAW codebook construction process; on the contrary, our hypothesis was that it might adversely affect the classification performance.

In this study we examined this stochasticity; we focused on measuring the variance caused by the randomness propagated to the next classification step. We found that, for three different paralinguistic datasets, this noise was responsible for a 3-8% absolute difference measured among identical models, differing in the BoAW random seed only. Furthermore, we found that the differences measured in cross-validation and on the test set were practically unrelated, confirming that this is indeed just the effect of stochasticity. We noted that this high variance, which was observed regardless of the actual codebook size used, makes model selection quite challenging.

In the next part of our study we noted the similarity between extracting the BoAW representation and applying random projection, since the feature representation of the examples is calculated in a non-deterministic way in both cases. To exploit this, we proposed to train an ensemble of classifiers; that is, we trained a separate SVM model for each BoAW variation, and fused their predictions by averaging out their posterior estimates. We showed that this way a significant performance increase

can be obtained: we achieved relative error reduction scores of 12-15% on two datasets, compared to relying on the BoAW model with the first random seed value, while on the third corpus we achieved a slight improvement (3%).

We presented our experimental results on three paralinguistic datasets, differing in speaker tasks, recording conditions and language as well as in the phenomenon we wish to detect. The proposed ensemble BoAW approach brought improvements in each case, which, in our view, confirms its general applicability, utility and robustness. Although the Bag-of-Audio-Words process has several hyperparameters, and it can also be used with various frame-level feature sets as input, we have no reason to suppose that the proposed ensemble BoAW technique does not bring significant improvements for other settings. Moreover, the proposed ensemble method could be applied on other, similarly stochastic utterance-level feature representations such as Fisher Vectors [5], [71]. This, however, is clearly the subject of future work.

REFERENCES

[1] S. L. Tóth, D. Sztahó, and K. Vicsi, "Speech emotion perception by human and machine," in *Proc. COST Action*, 2012, pp. 213–224.

[2] C. Busso, S. Mariooryad, A. Metallinou, and S. Narayanan, "Iterative feature normalization scheme for automatic emotion detection from speech," *IEEE Trans. Affect. Comput.*, vol. 4, no. 4, pp. 386–397, Oct.–Dec. 2013.

[3] H. Kaya and A. A. Karpov, "Efficient and effective strategies for cross-corpus acoustic emotion recognition," *Neurocomputing*, vol. 275, pp. 1028–1034, 2018.

[4] G. Paraskevopoulos, E. Tzinis, N. Ellinas, T. Giannakopoulos, and A. Potamianos, "Unsupervised low-rank representations for speech emotion recognition," in *Proc. Interspeech*, 2019, pp. 939–943.

[5] G. Gosztolya, "Using the Fisher Vector representation for audio-based emotion recognition," *Acta Polytechnica Hungarica*, vol. 17, no. 6, pp. 7–23, 2020.

[6] F. Grèzes, J. Richards, and A. Rosenberg, "Let me finish: Automatic conflict detection using speaker overlap," in *Proc. Interspeech*, 2013, pp. 200–204.

[7] H. Kaya, T. Özkaptan, A. A. Salah, and F. Gürgen, "Random discriminative projection based feature selection with application to conflict recognition," *IEEE Signal Process. Lett.*, vol. 22, no. 6, pp. 671–675, Jun. 2015.

[8] I. Hoffmann, D. Németh, C. D. Dye, M. Pákáski, T. Irinyi, and J. Kálmán, "Temporal parameters of spontaneous speech in Alzheimer's disease," *Int. J. Speech-Lang. Pathol.*, vol. 12, no. 1, pp. 29–34, 2010.

[9] J.-R. Orozco-Arroyave, J. D. Arias-Londoño, J. F. Vargas-Bonilla, and E. Nöth, "Analysis of speech from people with Parkinson's disease through nonlinear dynamics," in *Proc. NoLISP*, 2013, pp. 112–119.

[10] N. Garcia, J. C. Vásquez Correa, J. R. Orozco-Arroyave, and E. Nöth, "Multimodal i-vectors to detect and evaluate Parkinson's disease," in *Proc. Interspeech*, 2018, pp. 2349–2353.

[11] K. L. Fors, K. C. Fraser, and D. Kokkinakis, "Automated syntactic analysis of language abilities in persons with mild and subjective cognitive impairment." in *Proc. MIE*, 2018, pp. 705–709.

[12] J. Weiner and T. Schultz, "Selecting features for automatic screening for dementia based on speech," in *Proc. SPECOM*, 2018, pp. 747–756.

[13] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, May 2006.

[14] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[15] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Proc. SLT*, 2016, pp. 165–170.

[16] S. Amiriparian, M. Freitag, N. Cummins, and B. W. Schuller, "Sequence to sequence autoencoders for unsupervised representation learning from audio," in *Proc. DCASE*, 2017, pp. 17–21.

[17] B. W. Schuller *et al.*, "The INTERSPEECH 2018 computational paralinguistics challenge: Atypical & self-assessed affect, crying & heart beats," in *Proc. Interspeech*, Sep. 2018, pp. 122–126.

[18] S. Pancoast and M. Akbacak, "Bag-of-audio-words approach for multimedia event classification," in *Proc. Interspeech*, Sep. 2012, pp. 2105–2108.

[19] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[20] J. Han, Z. Zhang, M. Schmitt, Z. Ren, F. Ringeval, and B. Schuller, "Bags in bag: Generating context-aware bags for tracking emotions from speech," in *Proc. Interspeech*, 2018, pp. 3082–3086.

[21] F. B. Pokorny, F. Graf, F. Pernkopf, and B. W. Schuller, "Detection of negative emotions in speech signals using bags-of-audio-words," in *Proc. ACII*, Sep. 2015, pp. 1–5.

[22] M. Schmitt *et al.*, "A Bag-of-audio-words approach for snore sounds' excitation localisation," in *Proc. Speech Commun.*, Oct. 2016, pp. 89–96.

[23] H. Lim, M. J. Kim, and H. Kim, "Robust sound event classification using LBP-HOG based bag-of-audio-words feature representation," in *Proc. Interspeech*, 2015, pp. 3325–3329.

[24] S. Steidl, *Automatic Classification of Emotion-Related User States in Spontaneous Children's Speech*. Berlin, Germany: Logos Verlag, 2009.

[25] N. Cummins, M. Schmitt, S. Amiriparian, J. Krajewski, and B. Schuller, ""'You sound ill, take the day off'": Automatic recognition of speech affected by upper respiratory tract infection," in *Proc. EMBC*, Jul. 2017, pp. 3806–3809.

[26] S. Malmasi *et al.*, "A report on the 2017 native language identification shared task," in *Proc. BEA Workshop*, 2017, pp. 62–75.

[27] S. Rawat, P. F. Schulam, S. Burger, D. Ding, Y. Wang, and F. Metze, "Robust audio-codebooks for large-scale event detection in consumer videos," in *Proc. Interspeech*, 2013, pp. 2929–2933.

[28] 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 *Proc. Interspeech*, 2016, pp. 495–499.

[29] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. SODA*, Jan. 2007, pp. 1027–1035.

[30] E. Acar, F. Hopfgartner, and S. Albayrak, "Violence detection in Hollywood movies by the fusion of visual and mid-level audio cues," in *Proc. ACM Multimedia*, 2013, pp. 717–720.

[31] B. Schuller *et al.*, "The INTERSPEECH 2017 computational paralinguistics challenge: Addressee, cold & snoring," in *Pro. Interspeech*, 2017, pp. 3442–3446.

[32] N. Cummins, S. Amiriparian, S. Ottl, M. Gerczuk, M. Schmitt, and B. Schuller, "Multimodal bag-of-words for cross domains sentiment analysis," in *Proc. ICASSP*, 2018, pp. 4954–4958.

[33] Q. Jin *et al.*, "UCMM at MediaEval 2015 Affective impact of movies task: Fusion of audio and visual cues," in *Proc. MediaEval Workshop*, 2015, p. 26.

[34] P. S. Bradley and U. M. Fayyad, "Refining initial points for K-means clustering," in *Proc. ICML*, 1998, pp. 91–99.

[35] J. Grzybowska and S. Kacprzak, "Speaker age classification and regression using i-vectors," in *Proc. Interspeech*, Sep. 2016, pp. 1402–1406.

[36] A. Baird, E. Coutinho, J. Hirschberg, and B. W. Schuller, "Sincerity in acted speech: Presenting the sincere apology corpus and results," in *Proc. Interspeech*, 2019, pp. 539–543.

[37] S. Furui, *Speaker Recognit. in Smart Environ.* New York, NY, USA: Academic Press, 2010, ch. 7, pp. 163–184.

[38] Y. Muthusamy, E. Barnard, and R. Cole, "Reviewing automatic language identification," *IEEE Signal Process. Mag.*, vol. 11, no. 4, pp. 33–41, Oct. 1994.

[39] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Commun.*, vol. 35, pp. 115–124, 2001.

[40] H. Kwan and K. Hirose, "Use of recurrent network for unknown language rejection in language identification system," in *Proc. Eurospeech*, 1997, pp. 63–66.

[41] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proc. ACL*, 2005, pp. 515–522.

[42] A. Thyme-Gobbel and S. Hutchins, "On using prosodic cues in automatic language identification," in *Proc. ICSLP*, 1996, pp. 1768–1771.

[43] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. Interspeech*, 2011, pp. 857–860.

[44] E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Proc. ICASSP*, 2014, pp. 4080–4084.

[45] P. Ghahremani *et al.*, "End-to-end deep neural network age estimation," in *Proc. Interspeech*, 2018, pp. 277–281.

[46] M. Markitantov and O. Verkholyak, "Automatic recognition of speaker age and gender based on Deep Neural Networks," in *Proc. SPECOM*, 2019, pp. 327–336.

[47] R. Pappagari, T. Wang, J. Villalba, N. Chen, and N. Dehak, "X-vectors meet emotions: A study on dependencies between emotion and speaker recognition," in *Proc. ICASSP*, 2020, pp. 7169–7173.
[48] L. Moro-Velazquez, J. Villalba, and N. Dehak, "Using X-vectors to automatically detect Parkinson's disease from speech," in *Proc. ICASSP*, 2020, pp. 1155–1159.
[49] S. Hantke *et al.*, "I hear you eat and speak: Automatic recognition of eating condition and food type, use-cases, and impact on ASR performance," *PLoS One*, pp. 1–24, 2016.
[50] B. Schuller *et al.*, "The INTERSPEECH 2015 computational paralinguistics challenge: Nativeness, Parkinson's & eating condition," in *Proc. Interspeech*, 2015, pp. 478–482.
[51] D. Sztahó, V. Imre, and K. Vicsi, "Automatic classification of emotions in spontaneous speech," in *Proc. COST 2102*, 2011, pp. 229–239.
[52] K. Vicsi and D. Sztahó, "Recognition of emotions on the basis of different levels of speech segments," *J. Adv. Comput. Intell. Intell. Inform.*, vol. 16, no. 2, pp. 335–340, 2012.
[53] T. F. Yap, "Speech production under cognitive load: Effects and classification," Ph.D. dissertation, University of New South Wales, 2012.
[54] J. R. Stroop, "Studies of interference in serial verbal reactions," *J. Exp. Psychol.*, vol. 18, no. 6, pp. 643–662, 1935.
[55] B. Schuller *et al.*, "The INTERSPEECH 2014 computational paralinguistics challenge: Cognitive & physical load," in *Proc. Interspeech*, 2014, pp. 427–431.
[56] M. Schmitt and B. Schuller, "openXBOW – Introducing the Passau open-source crossmodal bag-of-words toolkit," *J. Mach. Learn. Res.*, vol. 18, pp. 1–5, 2017.
[57] S. Pancoast and M. Akbacak, "Softening quantization in bag-of-audio-words," in *Proc. ICASSP*, May 2014, pp. 1370–1374.
[58] H. Kaya, A. A. Karpov, and A. A. Salah, "Fisher vectors with cascaded normalization for paralinguistic analysis," in *Proc. Interspeech*, 2015, pp. 909–913.
[59] G. Gosztolya and L. Tóth, "A feature selection-based speaker clustering method for paralinguistic tasks," *Pattern Anal. Appl.*, vol. 21, no. 1, pp. 193–204, 2018.
[60] K. J. Han, S. Kim, and S. S. Narayanan, "Strategies to improve the robustness of agglomerative hierarchical clustering under data source variation for speaker diarization," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 16, no. 8, pp. 1590–1601, Nov. 2008.
[61] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 1–27, 2011.
[62] B. W. Schuller *et al.*, "The interspeech 2013 computational paralinguistics challenge: Social signals, conflict, emotion, autism," in *Proc. Interspeech*, Sep. 2013, pp. 148–152.
[63] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: The munich versatile and fast open-source audio feature extractor," in *Proc. ACM Multimedia*, 2010, pp. 1459–1462.
[64] G. Gosztolya, T. Grósz, R. Busa-Fekete, and L. Tóth, "Determining native language and deception using phonetic features and classifier combination," in *Proc. Interspeech*, Sep. 2016, pp. 2418–2422.
[65] D. Tavarez *et al.*, "Exploring fusion methods and feature space for the classification of paralinguistic information," in *Proc. Interspeech*, Aug. 2017, pp. 3517–3521.
[66] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.
[67] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, 1999.
[68] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.
[69] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. ACM SIGKDD*, Aug. 2001, pp. 245–250.
[70] P. Li, T. J. Hastie, and K. W. Church, "Very sparse random projections," in *Proc. ACM SIGKDD*, Aug. 2006, pp. 287–296.
[71] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. NIPS*, 1998, pp. 487–493.

Gábor Gosztolya received the M.Sc. degree in computer science and the Ph.D. degree in speech recognition from the University of Szeged, Hungary. He is a Senior Research Scientist with the Research Group on Artificial Intelligence of the Hungarian Academy of Sciences in Szeged. His research interests include speech recognition, computational paralinguistics, medical speech processing applications and applied machine learning.

Róbert Busa-Fekete received the M.Sc. degree in computer science/mathematics and the Ph.D. degree in machine learning from the University of Szeged, Hungary, in 2004 and 2009, respectively. He is a Research Scientist with Google Research. Between 2009 and 2012, he was a Postdoctoral Fellow with the CNRS/University of Paris-Sud, France, where he worked on various aspects of boosting algorithms and learning to rank problems. Prior to joining to Google Research, he worked as a Research Scientist with the Scalable Machine Learning group of Yahoo Research, NY, USA. His research span the field of Machine learning, in particular, online learning, multi-label classification, preference learning and their applications in other fields, such as online advertisement, speech recognition, natural language processing and bioinformatics.