Interspeech 2024
1-5 September 2024, Kos, Greece

Combining Acoustic Feature Sets for Detecting Mild Cognitive Impairment in the Interspeech'24 TAUKADIAL Challenge

Gábor Gosztolya[1,2], *László Tóth*[2]

[1]HUN-REN–SZTE Research Group on Artificial Intelligence, Szeged, Hungary
[2]University of Szeged, Institute of Informatics, Szeged, Hungary
{ ggabor, tothl } @ inf.u-szeged.hu

Abstract

Shared tasks or challenges provide valuable opportunities for the machine learning community, as they offer a chance to compare the performance of machine learning approaches without peeking (due to the hidden test set). We present the approach of our team for the Interspeech'24 TAUKADIAL Challenge, where the task is to distinguish patients of Mild Cognitive Impairment (MCI) from healthy controls based on their speech. Our workflow focuses entirely on the acoustics, mixing standard feature sets (ComParE functionals and wav2vec2 embeddings) and custom attributes focusing on the amount of silent and filled pause segments. By training dedicated SVM classifiers on the three speech tasks and combining the predictions over the different speech tasks and feature sets, we obtained F1 values of up to 0.76 for the MCI identification task using cross-validation, while our RMSE scores for the MMSE estimation task were as low as 2.769 (cross-validation) and 2.608 (test).
Index Terms: Mild Cognitive Impairment, wav2vec 2.0, shared task, Taukadial Challenge

1. Introduction

Mild cognitive impairment (MCI) is often considered to be a precursor of Alzheimer's disease (AD), although it may be a sign of other neurodegenerative diseases as well [1]. MCI may be present up to 15 years before a clear clinical manifestation of dementia, providing a wide time range to reduce the rate of cognitive decline [2]. The prevalence of MCI ranges from 15% to 20% in individuals of 60 years and older, and the annual progression rate from MCI to dementia is between 8% and 15% [3]. MCI affects several memory-related domains, such as the language skills, so the changes in language (and speech) performance may be early indicators of MCI [4].

This study describes our system for the Interspeech'24 TAUKADIAL Challenge (see [5]). Here, the task is to detect MCI based on the speech of English and Chinese subjects describing the contents of pictures. Challenges and shared tasks are valuable opportunities for the machine learning community, since they offer a chance to evaluate different machine learning approaches without peeking due to using a hidden test set and a limited number of submissions [6].

Our study focuses on the acoustic signal and ignores word-level information entirely for two reasons. Firstly, our interest lies in the acoustic processing of speech signals, and we believe that acoustic-only workflows deserve attention even if omitting Natural Language Processing (NLP) techniques might lead to some performance loss. Secondly, approaches focusing only on the acoustics of speech are likely to be more language-independent than those which utilize NLP tools (e.g. word embeddings [7, 8]). This latter reason is even more apparent in the case of the TAUKADIAL challenge, as here the speech utterances are either in English or in Chinese [5], making NLP methods more difficult to apply than in a single-language case.

Our system utilizes four feature extraction approaches. The first one is the 'ComParE functionals' feature set, which evolved into a widely-employed solution for computational paralinguistics, and was utilized, among others, in tasks such as estimating speaker age [9], sincerity [10] and determining whether the speaker has a cold [11]. The second approach involved using a DNN-based technique to summarize the amount of pause present in the actual utterance [12]. As the third and fourth techniques, we used embeddings from different blocks of a wav2vec 2.0 neural network [13]. Besides employing these methods on their own, we also employed combinations of the four methods.

2. Data

The acoustic data of the Interspeech'24 TAUKADIAL Challenge contains the speech of 169 (English and Chinese) subjects describing the content of three pictures [5]. From the 169 subjects, 129 appear in the training set, while 40 consitute the hidden test set, giving a total of 387 and 120 utterances, respectively. The two machine learning tasks were 1) to distinguish the MCI and normal control (NC) subjects, and 2) to estimate the Mini-Mental State Examination (MMSE) score of the subjects. The first one was a binary classification task, where the official evaluation metric was the Unweighted Average Recall (UAR) score, while in the second, regression task the metric to be minimized was the Root-Mean Square Error (RMSE). (Unfortunately, our team made the mistake of optimizing the classification models for the F_1 metric; this is elaborated further in the Conclusions and discussion section (see Section 7).) Up to five predictions could be submitted at the same time.

3. Experimental setup

In our experiments we employed the same machine learning workflow for all the feature sets tested, focusing on developing a procedure to set all the metaparameters as robustly as possible even with limited data (i.e. low number of subjects), this being typical in the pathological speech processing area. Next, we shall discuss the steps we applied in our workflow.

3.1. Preprocessing

Firstly, all the utterances were converted to a 16 kHz sampling rate monochannel format with a 16-bit resolution (surprisingly, not all recordings were distributed in this format). Furthermore, as the volume of the recordings varied greatly (perhaps mirroring the typical clinical environments), we applied automatic volume normalization.

10.21437/Interspeech.2024-984

3.2. Classification

We applied Support Vector Machines (SVM) for classification in the MCI-HC categorization task, and we used Support Vector Regression (SVR) to estimate the MMSE scores. For these, used the LibSVM [14] library. We employed the nu-SVM / nu-SVR method with linear kernel for the sake of robustness, based on our previous experiences in pathological speech processing and paralinguistic tasks [12, 15].

3.3. Metaparameter optimization

The value of the SVM C meta-parameter was tested in the range $10^{-5}, 10^{-4}, \ldots, 10^{1}$, determined via cross-validation over the whole training set. For this, we applied 20-fold cross-validation, where each fold consisted of the utterances of both HC and MCI subjects. To increase model robustness, we repeated the experiments with 5 different fold assignments. In the results sections, we will always report the mean of the five metric scores (for cross-validation). Final classifier models were trained using the optimal C value found this way, using all the examples of the whole training set. (Since there was no random element involved in final model training, there was no need to train multiple models.)

3.4. Evaluation

Performance for the **classification** task was measured using the UAR score, but (due to our mistake) we performed (and report) model selection based on the MCI task's F_1 score. (We will also report the Area Under the ROC Curve (AUC) values, which are also commonly applied in pathological speech processing studies [16].) Regarding the **regression** experiments, besides RMSE of the MMSE predictions, we also report the Pearson's correlation coefficient [17]. Before evaluation, the predicted MMSE scores were rounded to the nearest integer value, which also gave a slight improvement in our preliminary tests.

3.5. Prediction fusion

Following our preliminary experiments, during classification we treated each speech task separately; that is, we trained separate SVM / SVR modes for the recordings ending with '`-1`', '`-2`' and '`-3`'. To allow for a better modelling of each speech task, we tuned the metaparameters individually for each speech task, following the cross-validation procedure described in Section 3.2. Since the task was to predict the speaker category (MCI / HC) or the MMSE value of the *subject*, in the next step we fused the predictions obtained for the three speech tasks. When performing classification, we took the average of the posterior scores, while for the regression task it was quite straightforward to take the mean of the predicted MMSE values for the three speech tasks. To avoid overfitting and to increase model robustness, we decided not to assign weights for the speech tasks, but to perform this averaging in an unweighted (i.e. equivalently weighted) manner.

In later experiments we also fused the predictions obtained over the various feature sets. The actual procedure for choosing the feature sets to combine will be described in Section 5.1, but prediction fusion was performed in a similar way as that described previously: by taking the mean of the posterior estimates (classification) or the MMSE predictions (regression), and by using equal weights to improve model robustness.

3.6. Utterance chunking

In many speech processing tasks it is common to process the audio in short(er), equal-sized chunks. We also decided to experiment with this technique. We used 30 seconds-long utterance chunks with 50% overlap (i.e. a shift value of 15 seconds); the minimal length of a chunk was 10 seconds. During cross-validation, all the chunks of one speaker were assigned to the same fold . The predictions obtained for the chunks were merged in an unweighted manner (similarly to the procedure described in Section 3.5) to get subject-level scores, so evaluation was still done at the subject level.

4. Feature extraction

Next, we will describe the acoustic feature extraction methods we utilized.

4.1. ComParE functionals

As the first utterance-level feature extraction approach, we used the 'ComParE functionals' developed by Schuller et al [18]. The feature set includes energy, spectral, cepstral (MFCC) and voicing related frame-level features, from which specific functionals (like the mean, standard deviation, 1st and 99th percentiles or peak statistics) are computed to provide 6373 utterance-level feature values. This feature set was extracted by using the python port of the openSMILE tool [19].

4.2. Pause statistics

Our second method used seeks to quantify the amount of pause present in a given recording, which phenomenon is known to be relevant in acoustic MCI detection [4, 12, 20]. In general, we distinguish two types of pauses: silent and filled pauses, where the latter correspond to vocalizations like 'um', 'uh', 'er' etc. The method used consisted of the following steps:

(i) A standard Deep Neural Network acoustic model (from a HMM/DNN hybrid model) was evaluated on the actual utterance, using frame-level features (e.g. MFCCs).

(ii) Based on the output provided by the DNN, we estimated the local posterior probability values of silence and filler events. This step is still performed at the frame level.

(iii) From the local posterior estimates calculated in step (ii), new representations were computed at the utterance level by calculating the ratio of frames where the posterior estimate of a pause type exceeded a given threshold.

Step (iii) was performed with thresholds between 0 and 1 with 0.02 increments, either for the silent pauses, for the filled pauses, and for both pause types (i.e. adding up the posterior estimates of the two pauses), resulting in a total of $3 \times 50 = 150$ features for each utterance.

The acoustic DNN (of step (i)) was trained on a subset of the BEA Hungarian Spontaneous Speech Dataset [21]: 60 hours of data from 165 speakers were 'augmented' by adding noise, background babble and reverberation to the utterances, increasing the total amount of training data to 240 hours. Training was done on log-energies extracted from a Mel-scale filter bank (using a sliding window 150 ms), including their first and second order derivatives (i.e. Δ and $\Delta\Delta$). The network had five hidden layers of 1024 ReLU neurons in each, while the output layer contained 911 neurons (corresponding to context-dependent phonetic targets) with a softmax activation.

Table 1: *F_1 and AUC values obtained via cross-validation in the classification experiments. Approaches marked with an asterisk ('*') were evaluated on the test set*

Feature set	Chunks	F_1	AUC
ComParE functionals *	—	71.52%	0.677
	yes	72.73%	0.668
Pause statistics	—	61.87%	0.605
	yes	54.84%	0.643
wav2vec 2.0 (Conv.) *	—	73.55%	0.721
	yes	73.20%	0.698
wav2vec 2.0 (Fine-tuned)	—	73.08%	0.655
	yes	70.20%	0.681

Table 2: *Correlation and RMSE values obtained via cross-validation in the regression experiments. Approaches marked with an asterisk ('*') were evaluated on the test set*

Feature set	Chunks	Corr.	RMSE
ComParE functionals	—	0.455	2.971
	yes	0.404	3.079
Pause statistics *	—	0.525	2.887
	yes	0.489	2.951
wav2vec 2.0 (Conv.)	—	0.421	3.054
	yes	0.444	2.988
wav2vec 2.0 (Fine-tuned) *	—	0.533	2.833
	yes	0.479	2.932

4.3. wav2vec 2.0

wav2vec is basically a convolutional neural network (CNN) designed to process raw audio signals as input and generate representations suitable for automatic speech recognition (ASR) systems. The model is trained in a self-supervised manner, where it learns to predict future observations for the given speech sample [22]. This self-supervised training allows the model to be pre-trained on large, unannotated corpora, enabling subsequent fine-tuning for specific audio processing tasks such as ASR for low-resource languages [23] or paralinguistic appications (e.g. emotion detection [24]). The **wav2vec 2.0** architecture further enhances this approach by incorporating masking during training. Specifically, raw audio is encoded using a block of convolutional neural networks, and small segments of the resulting latent speech representations are masked, akin to masked language modeling. These masked representations are then processed by a quantizer, which selects speech units from an inventory of learned units, and a transformer network, which incorporates information from the entire utterance [13].

The outputs from the multi-layer convolutional block consist of the sequence of extracted feature vectors of the last convolutional layer, while the outputs from the second (fine-tuned) block comprise the sequence of the neural activations of the last layer in the block. These two types of feature vectors may carry relevant information for a large range of speech processing tasks: the former vector can be expected to capture lower-level information (e.g. pause-related information), while the fine-tuned layer can be expected to store phonetic-related information; so these vectors could be used as features [25, 26]. As the number of these (frame-level) feature vectors is proportional to the length of the utterance, they have to be aggregated over the whole recording. To do this, taking the mean and/or the standard deviation of the values over the whole utterance is a generally accepted solution [27, 28, 29].

Here, we used the wav2vec 2.0 model `wav2vec2-large-xlsr53-english`. The base of this model is the `XLSR-53` model pre-trained by Facebook on the audio data of 53 languages simultaneously [30]. This base model was then fine-tuned by `jonatasgrosman` [31] on the English part of the Mozilla Common Voice 6.1 corpus (2182 hours). The last layer of the convolutional block of this model consists of 512 neurons, while the last layer of the fine-tuned block has 1024 neurons. By using the mean and standard deviation values of these frame-level embedding vectors, we obtained 1024 and 2048 utterance-level features, convolutional and fine-tuned embeddings, respectively.

5. Cross-validation results

Table 1 shows the results obtained for the classification experiments. Due to the high number of possible cases, we present the values obtained on the level of subjects, i.e. after fusing the predictions for the three speech tasks. In general, we can see that by using the ComParE functionals features or those derived from the wav2vec 2.0 model (either from the convolutional or the fine-tuned block), we could obtain F_1 scores slightly above 70%, while the AUC scores were between 0.655 and 0.721. Using the pause statistics as attributes, however, was significantly less efficient, with F_1 values below 62% and AUC scores in the range $0.605 \ldots 0.643$. Splitting the utterances into chunks both for training and evaluation was not really efficient: it led to a slight improvement in the F_1 score only in one case out of four, while in two cases the AUC values were slightly better.

In the regression case (Table 2.) the observed trends are quite different: the best results were obtained via the fine-tuned wav2vec 2.0 embeddings, followed closely by the pause statistics attributes (correlation coefficients between 0.479 and 0.533, and RMSE values between 2.833 and 2.951). With the remaining two feature sets we obtained similar, but notably worse scores: correlation coefficients between 0.404 and 0.455, and RMSE scores between 2.971 and 3.079. Using utterance chunking did not really help either: it helped slightly for the wav2vec 2.0 convolutional attributes, but made the predictions worse in the remaining three cases. Due to these findings, we will not report any further results obtained via utterance chunking.

5.1. Feature set combinations

Besides using the four feature extraction methods separately, we also sought to improve classification and regression performance by combining them. Combination was always carried out by *late fusion*, i.e. fusing the predictions obtained by the separate feature sets. To select the best feature set combination, we used an approach similar to the sequential forward feature selection technique [32, 33]: first we took the feature set which led to the highest classification (see Table 1.) or regression (see Table 2.) performance. Then we tried adding each remaining feature set, and chose the combination which led to the best performance. We repeated this procedure until all the feature sets were combined. (Recall that we report only the results obtained over the full recordings.)

The results of this combination procedure for the **classification** task can be seen in Table 3.; the best values (and those falling close to them) are shown in **bold**. In addition to the performance of the standalone wav2vec 2.0 convolutional embed-

Table 3: F_1 *and AUC values obtained via cross-validation in the* **classification** *experiments with method combinations (using the* **full recordings***). Approaches marked with an asterisk ('*') were evaluated on the test set*

Feature sets	F_1	**AUC**
wav2vec2 (C.) + ComParE func. *	**76.43**	**0.726**
wav2vec2 (C.) + Pause statistics	70.59	0.710
wav2vec2 (C.) + wav2vec2 (F.) *	**76.25**	**0.733**
w2v2 (C.) + ComParE f. + w2v2 (F.) *	75.31	**0.729**
w2v2 (C.) + ComParE f. + Pause s.	75.32	0.720
All four methods	75.16	0.725

Table 4: *Correlation coefficients and RMSE values obtained via cross-validation in the* **regression** *experiments with method combinations (using the* **full recordings***). Approaches marked with an asterisk ('*') were evaluated on the test set*

Feature sets	**Corr.**	**RMSE**
wav2vec2 (F.) + ComParE func.	0.532	2.839
wav2vec2 (F.) + Pause statistics *	**0.597**	**2.769**
wav2vec2 (F.) + wav2vec2 (C.)	0.514	2.914
w2v2 (F.) + Pause s. + ComParE f. *	**0.585**	**2.790**
w2v2 (F.) + Pause s. + w2v2 (C.)	0.565	2.839
All four methods *	**0.586**	2.812

dings feature set (F_1 score of 73.55%), notable improvements were achieved by combination: by including either the ComParE functionals attributes or the fine-tuned wav2vec 2.0 embeddings, the F_1 value increased over 76%, and the AUC score increased slightly as well. However, three-wise or four-wise combinations did not help any further, although these combined predictions slightly outperformed all standalone methods.

The combined results for the **regression** task can be seen in Table 4; the best values (and those close to them) are again shown as **bold**. In this case the best performance was achieved by combining the two best standalone methods, i.e. fine-tuned wav2vec 2.0 embeddings and pause statistics. Including any further attributes led to somewhat worse scores, although using all four methods was still better than any standalone method.

6. Test set results

The top half of Table 5. shows the scores obtained on the test set, expressed in F_1 and UAR. Overall, the results are around chance level, which we will discuss more in detail in Section 7. (The main reason is probably that our team optimized for the wrong metric (F_1 instead of UAR) in the classification task.) Regarding the regression experiments (see the bottom half of Table 5.), our submissions were more successful. Just as in cross-validation, we obtained the lowest RMSE and highest correlation values with pause statistics, fine-tuned wav2vec2 embeddings and their combinations. Also notice that the range of RMSE values (2.608. . .2.702) was quite similar to those obtained in the cross-validation setting (2.769. . .2.887). This, in our opinion, confirms the robustness of our meta-parameter setting procedure, at least for the regression experiments (although the correlation values are slightly lower (0.400. . .0.457) than they were in cross-validation (0.525. . .0.597)). Based on the baseline paper published in the last moments (see [5]), our predictions scores are quite competitive, as all of them is lower than the baseline 2.89 score (i.e. the RMSE score of the linguistic features) on the test set.

Table 5: F_1 *and AUC values obtained on the hidden test set.*

Classification	F_1	**UAR**
ComParE functionals	52.2	44.4%
wav2vec 2.0 (Conv.)	51.2	47.2%
wav2vec2 (Conv.) + ComParE func.	56.5	49.4%
wav2vec2 (Conv.) + wav2vec2 (F.)	56.5	49.4%
w2v2 (C.) + ComParE f. + w2v2 (F.)	52.2	44.1%
Regression	**Corr.**	**RMSE**
Pause statistics	0.439	2.608
wav2vec 2.0 (Fine-tuned)	0.457	2.660
wav2vec2 (F.) + Pause statistics	0.455	2.612
w2v2 (F.) + Pause s. + ComParE f.	0.400	2.702
All four methods	0.407	2.683

7. Conclusions and discussion

In this study we presented four feature extraction approaches applied for MCI detection in the Interspeech'24 TAUKADIAL Challenge. Due to the bilingual nature of the Challenge (i.e. the recordings were either in English or in Chinese), we employed acoustic features, which we expected to be more robust than NLP techniques. The two machine learning tasks in the Challenge were separating MCI and normal control subjects (a classification task), and predicting the MMSE values of the subjects (a regression task). We tested individual approaches along with combinations. Our results on the hidden test set were mixed: all five of our submissions resulted in chance-level predictions in the classification task, while for regression we obtained scores similar to those got via cross-validation, and better than those reported in the baseline paper.

In our opinion, our poor test set results in the classification task might be due to three reasons. Firstly, by mistake, we optimized for F_1, while the official ranking metric in this first task was UAR. The second reason is that F_1 is known to be a metric hard to optimize [34], which might explain why our classification attempts lacked robustness. A third option might simply be the specific properties of the actual data set, as the classification results reported in the baseline paper also lacked any robustness [5]. (Of course, a bug in one of our classification-related scripts cannot be ruled out either.)

In contrast, our regression methods turned out to be quite robust. This, in our opinion, validates our meta-parameter determination procedure, namely using several cross-validation loops to maximize robustness during selecting C for SVM and combining methods in an unweighted manner to avoid overfitting. Also note that, although one might expect the same features to be useful for MCI identification as those for MMSE estimation, we chose sharply different methods – ComParE functionals and convolutional embeddings for classification, while pause statistics and the fine-tuned embeddings for regression. As the former two methods were not exactly effective for detecting MCI, while the latter two attribute sets were robust for MMSE estimation (judging from test set results), it may be that the fine-tuned embeddings and pause statistics would have worked well on the hidden test set in the classification task as well. Of course, this should be investigated more thoroughly, which could be a subject for future works.

8. Acknowledgements

This study was supported by the NRDI Office of the Hungarian Ministry of Innovation and Technology (grant no. TKP2021-NVA-09), and within the framework of the Artificial Intelligence National Laboratory Program (RRF-2.3.1-21-2022-00004).

9. References

[1] R. C. Petersen, Ed., *Mild cognitive impairment: Aging to Alzheimer's disease*. Oxford University Press, 2003.

[2] E. A. Hahn and R. Andel, "Nonpharmacological therapies for behavioral and cognitive symptoms of mild cognitive impairment," *Journal of Aging and Health*, vol. 23, no. 8, pp. 1223–1245, 2011.

[3] R. C. Petersen, "Mild cognitive impairment," *Continuum: Lifelong Learning in Neurology*, vol. 22, no. 2 (Dementia), pp. 404–418, 2016.

[4] L. Tóth, I. Hoffmann, G. Gosztolya, V. Vincze, G. Szatlóczki, Z. Bánréti, M. Pákáski, and J. Kálmán, "A speech recognition-based solution for the automatic detection of mild cognitive impairment from spontaneous speech," *Current Alzheimer Research*, vol. 15, no. 2, pp. 130–138, 2018.

[5] S. Luz, S. d. l. F. Garcia, F. Haider, D. Fromm, B. MacWhinney, A. Lanzi, Y.-N. Chang, C.-J. Chou, and Y.-C. Liu, "Speech-based cognitive assessment in Chinese and English: The TAUKADIAL challenge," in *Interspeech*, 2024, p. to appear.

[6] M. Nissim, L. Abzianidze, K. Evang, R. van der Goot, H. Haagsma, B. Plank, and M. Wieling, "Sharing is caring: The future of shared tasks," *Computational Linguistics*, vol. 43, no. 4, pp. 897–904, 2017.

[7] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, and M. McDermott, "Publicly available clinical BERT embeddings," in *Proceedings of ClinicalNLP*, 2024, pp. 72–78.

[8] G. Berend, "Sparse coding of neural word embeddings for multilingual sequence labeling," *Transactions of the Association for Computational Linguistics*, vol. 2017, no. 5, pp. 247–261, 2017.

[9] J. Grzybowska and S. Kacprzak, "Speaker age classification and regression using i-vectors," in *Proceedings of Interspeech*, San Francisco, CA, USA, Sep 2016, pp. 1402–1406.

[10] 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, CA, USA, 2016, pp. 2001–2005.

[11] B. Schuller, S. Steidl, A. Batliner, S. Hantke, E. Bergelson, J. Krajewski, C. Janott, A. Amatuni, M. Casillas, A. Seidl, M. Soderstrom, A. S. 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*, Aug 2017, pp. 3442–3446.

[12] J. V. Egas-López, R. Balogh, N. Imre, I. Hoffmann, M. K. Szabó, L. Tóth, M. Pákáski, J. Kálmán, and G. Gosztolya, "Automatic screening of mild cognitive impairment and Alzheimer's disease by means of posterior-thresholding hesitation representation," *Computer, Speech & Language*, vol. 75, no. Sep, 2022.

[13] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[14] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[15] J. V. Egas-López, M. Vetráb, L. Tóth, and G. Gosztolya, "Identifying conflict escalation and primates by using ensemble x-vectors and Fisher vector features," in *Proceedings of Interspeech*, Brno, Czech Republic, Aug 2021, pp. 476–480.

[16] L. Hason and S. Krishnan, "Spontaneous speech feature analysis for Alzheimer's disease screening using a random forest classifier," *Frontiers in Digital Health*, vol. 2022, no. 4, 2022.

[17] W. Kirch, "Pearson's correlation coefficient," in *Encyclopedia of Public Health*. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091.

[18] B. W. Schuller, S. Steidl, A. Batliner, A. Vinciarelli, K. Scherer, F. Ringeval, M. Chetouani, F. Weninger, F. Eyben, E. Marchi, H. Salamin, A. Polychroniou, F. Valente, and S. Kim, "The Interspeech 2013 Computational Paralinguistics Challenge: Social signals, Conflict, Emotion, Autism," in *Proceedings of Interspeech*, Lyon, France, Sep 2013, pp. 148–152.

[19] F. Eyben, M. Wöllmer, and B. Schuller, "OpenSMILE: The Munich versatile and fast open-source audio feature extractor," in *Proceedings of ACM Multimedia*, 2010, pp. 1459–1462.

[20] R. Balogh, N. Imre, G. Gosztolya, I. Hoffmann, M. Pákáski, and J. Kálmán, "The role of silence in verbal fluency tasks – a new approach for the detection of mild cognitive impairment," *Journal of the International Neuropsychological Society*, vol. 29, no. 1, pp. 46–58, 2023.

[21] T. Neuberger, D. Gyarmathy, T. Gráczi, V. Horváth, M. Gósy, and A. Beke, "Development of a large spontaneous speech database of agglutinative Hungarian language," in *TSD*, 2014, pp. 424–431.

[22] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proceedings of Interspeech*, 2019, pp. 3465–3469.

[23] P. Mihajlik, A. Balog, T. E. Gráczi, A. Kohári, B. Tarján, and K. Mády, "BEA-Base: A benchmark for ASR of spontaneous Hungarian," in *Proceedings of LREC*, Marseille, France, 2022, pp. 1970–1977.

[24] L. Pepino, P. Riera, and L. Ferrer, "Emotion recognition from speech using wav2vec 2.0 embeddings," in *Proceedings of Interspeech*, Brno, Czechia, 2021, pp. 3400–3404.

[25] W.-W. Lin and M.-W. Mak, "Wav2spk: A simple DNN architecture for learning speaker embeddings from waveforms," in *Proceedings of Interspeech*, 2020, pp. 3211–3215.

[26] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," in *Proceedings of Interspeech*, 2021, pp. 1509–1513.

[27] P. A. Pérez-Toro, P. Klumpp, A. Hernandez, T. Arias, P. Lillo, A. Slachevsky, A. M. García, M. Schuster, A. K. Maier, E. Nöth, and J. R. Orozco-Arroyave, "Alzheimer's detection from English to Spanish using acoustic and linguistic embeddings," in *Interspeech*, Incheon, Dél-Korea, 2022, pp. 2483–2487.

[28] G. Gosztolya, L. Tóth, V. Svindt, J. Bóna, and I. Hoffmann, "Using acoustic Deep Neural Network embeddings to detect Multiple Sclerosis from speech," in *Proceedings of ICASSP*, Singapore, May 2022, pp. 6927–6931.

[29] N. Vaessen and D. A. Van Leeuwen, "Fine-tuning wav2vec2 for speaker recognition," in *Proceedings of ICASSP*, 2021, pp. 7967–7971.

[30] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proceedings of Interspeech*, 2022, pp. 2278–2282.

[31] J. Grosman, "Fine-tuned XLSR-53 large model for speech recognition in English," https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-english, 2021.

[32] M. Last, A. Kandel, and O. Maimon, "Information-theoretic algorithm for feature selection," *Pattern Recognition Letters*, vol. 22, no. 6-7, pp. 799–811, 2001.

[33] I. A. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods*, vol. 43, no. 1, pp. 3–31, 2000.

[34] P. M. Chinta, P. Balamurugan, S. Shevade, and M. N. Murty, "Optimizing F-measure with non-convex loss and sparse linear classifiers," in *Proceedings of IJCNN*, Dallas, TX, USA, 2013, pp. 1970–1977.