ABSTRACT
The ACM Multimedia 2022 Computational Paralinguistics Challenge addresses four different problems for the first time in a research competition under well-defined conditions: In the Vocalisations and Stuttering Sub-Challenges, a classification on human non-verbal vocalisations and speech has to be made; the Activity Sub-Challenge aims at beyond-audio human activity recognition from smartwatch sensor data; and in the Mosquitoes Sub-Challenge, mosquitoes need to be detected. We describe the Sub-Challenges, baseline feature extraction, and classifiers based on the ‘usual’ ComPaRE and BoAW features, the auDeep toolkit, and deep feature extraction from pre-trained CNNs using the DeepSpectrum toolkit; in addition, we add end-to-end sequential modelling, and a LOG-MEL-128-BNN.

CCS CONCEPTS
- Information systems → Multimedia and multimodal retrieval, Computing methodologies → Artificial intelligence.

KEYWORDS
Computational Paralinguistics; Vocalisations; Stuttering; Human Activity Recognition; Mosquito Detection; Challenge; Benchmark

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference Format:
Björn Schuller, Anton Batliner, Shahin Amiriparian, Christian Bergler, Maurice Gerczuk, Natalie Holz, Pauline Larrouy-Maestri, Sebastian Bayerl, Korbinian Riedhammer, Ivan Kiskin, Marianne Sinka, and Stephen Roberts. 2022. The ACM Multimedia 2022 Computational Paralinguistics Challenge: Vocalisations, Stuttering, Activity, & Mosquitoes. In Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3503161.3551591

1 INTRODUCTION
In this ACM Multimedia 2022 Computational PARALINGUISTICS CHALLENGE (ComPaRE) – the 14th since 2009 [30], we address four new problems within the field of Computational Paralinguistics [29] in a challenge setting:

In the Vocalisations Sub-Challenge, non-verbal vocal expressions from the Variably Intense Vocalizations of Affect and Emotion Corpus [14, 15] are used (VOC-C) for classifying the expression of six different emotions. Such human non-verbals are still understudied but are ubiquitous in human communication [26].

In the Activity Sub-Challenge, parts (KSF-C) of the Kassel State of Fluency corpus [8, 9] are used. Stuttering is a complex speech disorder with a crude prevalence of about 1 % of the population [36]. Monitoring of stuttering would allow objective feedback to persons who stutter (PWS) and speech therapists, thus facilitating tailored speech therapy, with the automatic detection of different stuttering phenomena as a necessary prerequisite.

The human activity recognition corpus harAGE as used in the Activity Sub-Challenge (HAR-C), provided by the EU Horizon 2020 project sustAGE [21], is a multimodal dataset collected with the smartwatch Garmin Vivoactive 3 [19, 20]. The monitoring of
different types of physical activity vs inactivity is of vital importance to promote healthier and active life styles in the population, improving their overall physical health and wellbeing [12, 25].

The Mosquito corpus as used in the Mosquitoes Sub-Challenge (MOS-C), provided by the HumBug Project, is a large-scale audio database consisting of over 20 hours of mosquito flight recordings (HumBugDB [17]). Mosquitoes are responsible for more human deaths than any other creature; e.g., in 2020 malaria caused around 241 million cases of disease across more than 100 countries resulting in an estimated 627,000 deaths [23]. It is imperative therefore to accurately locate and identify dangerous mosquitoes to achieve efficient mosquito control.

For all tasks, a target class has to be predicted for each case. Contributors can employ their own features and machine learning (ML) algorithms; standard feature sets and procedures are provided. Participants have to use the pre-defined partitions for each Sub-Challenge. They may report results that they obtain from the Train(ing)/Dev(elopment) set but have only five trials to upload their results on the Test set per Sub-Challenge, whose labels are unknown to them. Each participation must be accompanied by a paper presenting the results, which undergoes peer-review. The organisers preserve the right to re-evaluate the findings, but will not participate in the Challenge. As evaluation measure, we employ for all Sub-Challenges but Mosquitoes the Unweighted Average Recall (UAR) as used since the first Challenge from 2009 [30, 31]; it is more adequate for (unbalanced) multi-class classifications than Weighted Average Recall (i.e., accuracy) [29]. The Mosquitoes Sub-Challenge is an audio event detection task; hence, we utilise the Polyphonic Sound Event Detection Score (PSDS) [10] – an extension for a classifier threshold-independent event-based F-score.

Ethical approval for the studies has been obtained.

2 THE FOUR SUB-CHALLENGES

Vocalisations – The Vocalisation Corpus VOC-C:
It is provided by the MPI for Empirical Aesthetics, Frankfurt am Main, featuring vocalisations – such as laughter, cries, moans, or screams – with different affective intensities, expressing different emotional states. The data from the female speakers have been made available to the public, see [14, 15]; the male speakers are so far unseen. We partition the female vocalisations into Train (6 speakers, 625 samples) and Dev(elopment) (5 speakers, 460 samples), and the male vocalisations (2 speakers, 276 samples) into Test, modelling a 6-class problem with the emotional classes achievement, anger, fear, pain, pleasure, and surprise.

Stuttering – The Kassel State of Fluency Corpus KSF-C:
The corpus provided by the TH Nürnberg and the Kasseler Stuttertherapie is derived from the Kassel State of Fluency (KSoF) corpus [8, 9]. The original corpus features 5,597 typical and nontypical (stuttering) 3 s segments from 37 German speakers with an overall duration of 4.6 h. The segments were annotated by three labellers as one of 7 classes (block, prolongation, sound repetition, word/phrase repetition, modified speech technique, interjection, no disfluency) and with some additional information, e.g., about the recording quality. Annotators were able to assign more than one label per segment. For this challenge, we removed all the ambiguously labelled segments, thus only featuring 4,601 segments. The task proposed in this challenge is the classification of speech segments as one of 8 classes – the seven stuttering-related classes and an eighth garbage class, denoting segments that are unintelligible, contain no speech, or are negatively affected by background noise. The dataset is split into three speaker-independent partitions (Train: 23 speakers, Dev: 6 speakers, Test: 8 speakers).

Activity – The Human Activity Recognition Corpus HAR-C:
The harAGE corpus [19, 20] contains 17 h 37 m 20 s of triaxial accelerometer, heart rate, and pedometer sensor measurements from 30 (14 f, 16 m) participants with a mean age of 40.0 years and a standard deviation of 8.3 years. Sensor measurements from eight activities are included: lying, sitting, standing, washing hands, walking, running, stairs climbing, and cycling. The dataset is split into three participant-independent and gender-balanced partitions. The Train, Dev, and Test partitions contain a total of 10 h 41 m 20 s, 2 h 16 m 0 s, and 4 h 40 m 0 s of data from 17 (8 f, 9 m), 6 (3 f, 3 m), and 7 (3 f, 4 m) participants, respectively. Each sample in the harAGE corpus contains 20 s of continuous sensor measurements from one participant performing one of the different activities considered in the dataset. The task in this Sub-challenge consists in the development of unimodal and/or multimodal systems able to analyse these 20 s of sensor measurements and infer the corresponding activity.

Mosquitoes – The Mosquito Corpus MOS-C:
It is provided by the HumBug Project and is strongly based on HumBugDB [17]. In a revision for this Sub-challenge, the former test set A is expanded with more challenging negatives and now forms Dev A. The former test set B forms Dev B, and the training set is identical. The task is to detect timestamps for acoustic mosquito events – Mosquito Event Detection (MED). The challenge is therefore scored in the time domain with the PSDS package [10]. Details of Train and Dev are given in [17, Sec. 4]. To summarise, Dev A represents semi-field conditions, where mosquitoes were manually released near recording setups that feature traditional housing constructions, equipped with mosquito bednets [35, Sec. 2.1.2]. Dev B is a low-SNR recording set of free-flying mosquitoes within culture cages. The data vary in sample rate, recording devices, ambient conditions, and experimental assumptions. As these factors can introduce confounding, they are given as metadata and documented in [17, Appx. C]. The test set consists of recordings conducted in South East Tanzania by volunteers in people’s homes. It is therefore not included in the hosted Zenodo dataset due to the sensitive nature of the data. Please note that participants will not receive the test data but will submit dockerised versions of their code using the help of the provided templates for either Tensorflow 2.0 [1], or PyTorch [24], that participants are free to choose as they wish.

3 EXPERIMENTS AND RESULTS

For the VOC-C and the KSF-C, the segmented audio was converted to single-channel 16 kHz, 16 bits PCM format. Table 1 shows the number of data for Train, Dev, and Test for the different corpora. MOS-C Dev was split into two sets with differing conditions.

1https://zenodo.org/record/6517688
2The full list of authors contributing to HumBugDB is in [17] and associated Zenodo repository
3v0.0.2 HumBugDB: https://zenodo.org/record/6478589
Table 1: Summary of the databases presented per Sub-Challenge. Number of instances per class in the Train/Dev/Test splits. Test split distributions were blinded during the ongoing challenge.

| Class      | VOC-C: classification task (%) | KSF-C: classification task (%) | HAR-C: classification task (%) | MOS-C: detection (in hours) |
|------------|--------------------------------|--------------------------------|--------------------------------|----------------------------|
| achiev.    | 89 72 48 209                   | Block                          | lyning                        | mosquito                   |
| anger      | 101 73 45 219                  | Fillers                        | sitting                       | non-mosquito                |
| fear       | 103 73 46 222                  | Garbage                        | standing                      | 13.4 2.7/0.56 17.1 33.8    |
| pain       | 114 71 47 232                  | Modified                       | wash hands                    | 133 35 60 228               |
| pleasure   | 109 93 43 245                  | Prolong                        | walking                       | 392 57 114 473              |
| surprise   | 109 78 47 234                  | SoundRep.                      | running                       | 301 40 98 439               |
|            |                                 | WordRep.                       | stairs climb                  | 263 43 110 416              |
|            |                                 | no_diff.                       | cycling                       | 186 58 105 349              |
| Σ          | 625 460 276 1361               | Σ                               | Σ                             | 30.4 3.8/0.81 18.0 53.0     |

Table 2: Results for the Sub-Challenges. The official baselines for Test are highlighted (bold and greyscale); there are no official baselines for Dev. UAR: Unweighted Average Recall. CI on Test: Confidence Intervals on Test, see explanation in the text. For the VOC-C and KSF-C, DenseNet169 and DenseNet201 are used to extract DeepSpectrum features, respectively.

| Approach     | Dev | Test | CI on Test | Dev | Test | CI on Test | Dev | Test | CI on Test |
|--------------|-----|------|------------|-----|------|------------|-----|------|------------|
| ComParE      |     |      |            |     |      |            |     |      |            |
| DeepSpectrum | 35.0| 34.1 | 29.5 – 39.2| 28.1| 33.7 | 33.5 – 41.4|     |      |            |
| auDeep       | 31.0| 31.2 | 26.1 – 36.6| 17.7| 25.9 | 21.9 – 30.3|     |      |            |
| BooWs        | 39.6| 37.4 | 32.6 – 42.8| 26.7| 32.1 | 28.2 – 36.0|     |      |            |
| Fusion       | 39.8| 36.1 | 31.3 – 31.3| 28.7| 38.3 | 34.3 – 41.9|     |      |            |

3.1 Approaches

**ComParE Acoustic Feature Set:** The official baseline feature set from openSMILE is the same as the one used in previous editions of the ComParE challenges, starting from 2013 [11, 32].

**DeepSpectrum:** It is applied to obtain deep representations from the input audio data utilising image pre-trained Convolutional Neural Networks (CNNs) [2, 4]. It has been used in previous challenges [33, 34] and is described in [4]. The efficacy of DeepSpectrum features have been demonstrated for speech and audio recognition tasks [6]. A lightweight version of DeepSpectrum for audio signal processing on-device can be found in [7].

**auDeep:** This feature set is obtained through unsupervised representation learning with recurrent sequence-to-sequence autoencoders [3, 13].

**Bag-of-Audio-Words (BoAWs):** Audio chunks are represented as histograms of ComParE Low-Level Descriptors (LLDs), after quantisation based on a codebook. They have been used in previous challenges [33, 34] and other studies [5, 18, 27]; the toolkit openXBOW is described in [28].

**End-to-end sequential modelling:** The HAR-C implements an end-to-end approach exploiting the sensor data as input. As described in [20], 3-dimensional, 2-dimensional, and 9-dimensional traces are generated from the raw heart rate, pedometer, and triaxial accelerometer measurements, respectively. As opposed to [20], herein, we do not debias the accelerometer measurements to ease the deployment of the presented approach in real-life applications.

The network implemented is composed of a dedicated feature extraction block for each modality – responsible for extracting deep learnt representations from the input traces – followed by a classification block – in charge of performing the actual inference. The feature extraction block implements a 1-dimensional convolutional layer, and the classification layer two fully connected layers. The dimensionality of the resulting features at the output of the feature extraction block depends on the number of modalities to be fused, concatenating the embedded representations learnt separately from each modality.

**log-mel-128-BNN:** MOS-C utilises a Bayesian Convolutional Neural Network with four convolutional, two max-pooling, and one fully connected layer, and the classification layer two fully connected layers. The dimensionality of the resulting features at the output of the feature extraction block depends on the number of modalities to be fused, concatenating the embedded representations learnt separately from each modality.

**3.2 Challenge Baselines and Interpretation**

We provide a branch on the official challenge repository for each Sub-Challenge, which includes scripts allowing participants to fully reproduce the baselines (including pre-processing, model training, and model evaluation on Dev). For VOC-C, KSF-C, and HAR-C, the 95% Confidence Intervals (CI) were computed by 1 000 bootstraping (random sampling with replacement) and UARs for Test.
based on the same model that was trained with Train and Dev. For MOS-C, as appropriate for sound event detection tasks, the 95% CIs are constructed using the jackknife method (leave-one-out sampling) [22] with the number of samples equal to the number of test audio recordings. Due to space restrictions, for VOC-C and KSF-C, we leave out the results for every hyperparameter configuration evaluated and only provide the best results obtained. The baselines for both VOC-C and KSF-C consist of using Support Vector Machines with linear kernels on four different audio feature representations – ComParE, DeepSpectrum, auDeep, and BoAWs. All feature representations are scaled to zero mean and unit standard deviation, using the parameters from the respective training set (when Train and Dev are fused for the final classifier, the parameters are calculated on this fusion). The SVM complexity parameter C is always optimised during the development phase.

**Vocalisations – VOC-C:** We obtain the best UAR=37.4 % on Test with BoAWs, see Table 2. Figure 1(a) shows, for the best Dev result given in Table 2, that the two classes pain and fear fall behind the other four classes, and that they are mostly confused with surprise.

**Stuttering – KSF-C:** We achieve UAR=40.4 % on Test with DeepSpectrum. Looking at the confusion matrix of our best result on Dev in Figure 1(b), word repetitions seem to be the hardest to detect and differentiate, especially from instances of modified speech and sound repetitions.

**Activity – HAR-C:** The best approach on Test fuses the heart rate, the pedometer, and the accelerometer modalities, scoring a UAR=72.2 %. The results highlight the importance of the accelerometer information for this task, as the models exploiting this modality outperform those using the heart rate and the pedometer information, either unimodally or multimodally. Analysing the confusion matrix given in Figure 1(c) for the best result on Dev in Table 2, we observe that the main confusions take place among the ‘non-moving’ activities lying, sitting, and standing.

**Mosquitoes – MOS-C:** Table 2 shows the PSDS of 61.4 % and 3.4 % achieved on Dev A and B, respectively. We note that the provided model is unable to achieve a good score on Dev B, which features a lower SNR, more challenging dataset. The baseline scores 12.2 % PSDS on the Test partition, which can be thought of as an approximate combination in recording conditions of Dev A and B. The results highlight the need to train a model that is able to perform well and generalise across different deployment scenarios. Each of the Dev A, Dev B, and Test sets features considerably different audio backgrounds, as they are recorded in different environments. Additional feature window-based metrics are supplied in the repository, which give a breakdown of performance by precision-recall, ROC, and confusion matrices. These may be helpful for developing with the ultimate aim of maximising the PSDS on Test.

## 4 CONCLUDING REMARKS

This year’s challenge is new by four new tasks, all of them highly relevant for applications. We feature our ‘classic’ approaches ComParE and Bag-of-Audio-Words (BoAWs), auDeep, and DeepSpectrum for VOC-C and KSF-C, and two new ones, tailored for HAR-C and MOS-C. For all computation steps, scripts are provided that can, but need not be used by the participants. We expect participants to obtain better performance measures by employing novel (combinations of) procedures and features, including such tailored to the particular tasks.

## 5 ACKNOWLEDGMENTS

We acknowledge funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No. 826506 (sustAGE), from the Deutsche Forschungsgemeinschaft (DFG) under grant agreement No. 421613952 (ParaStiChaD) as well as from the DFG’s Reinhart Koselleck project No. 44221874 (AUDIoNOMOUS), and from the Gates Foundation No. opp1209888, as well as the contributions of all authors in [17, HumBugDB] and the MOS-C Zenodo repository.
