Sound Event Detection in the DCASE 2017 Challenge

Annamaria Mesaros, *Member, IEEE*, Aleksandr Diment, Benjamin Elizalde, Toni Heittola, Emmanuel Vincent, *Senior Member, IEEE*, Bhiksha Raj, *Fellow, IEEE*, and Tuomas Virtanen, *Senior Member, IEEE*

***Abstract*—Each edition of the challenge on Detection and Classification of Acoustic Scenes and Events (DCASE) contained several tasks involving sound event detection in different setups. DCASE 2017 presented participants with three such tasks, each having specific datasets and detection requirements: Task 2, in which target sound events were very rare in both training and testing data, Task 3 having overlapping events annotated in real-life audio, and Task 4, in which only weakly labeled data were available for training. In this paper, we present three tasks, including the datasets and baseline systems, and analyze the challenge entries for each task. We observe the popularity of methods using deep neural networks, and the still widely used mel frequency-based representations, with only few approaches standing out as radically different. Analysis of the systems behavior reveals that task-specific optimization has a big role in producing good performance; however, often this optimization closely follows the ranking metric, and its maximization/minimization does not result in universally good performance. We also introduce the calculation of confidence intervals based on a jackknife resampling procedure to perform statistical analysis of the challenge results. The analysis indicates that while the 95% confidence intervals for many systems overlap, there are significant differences in performance between the top systems and the baseline for all tasks.**

***Index Terms*—Sound event detection, weak labels, pattern recognition, jackknife estimates, confidence intervals.**

I. Introduction

SOUND event detection is a considerably broad topic in the field of environmental sound detection and classification, with far-reaching applicability for surveillance and monitoring [1]–[3], assistive technologies [4]–[6], or multimedia indexing [7]. In this context, sound events are defined as individual sounds that convey information about what is happening in the scene, for example fire alarm, glass breaking, car horn, or dog barking, to name a few. The interest in recognizing sounds in audio recordings arises from the usefulness of such methods in the aforementioned applications. In many applications, the main purpose is to detect a small number of target sounds [8], [9], while a more generic research problem is to detect a large number of possibly overlapping sound events [3].

Everyday soundscapes can be described in terms of sounds at different levels, with the acoustic scene being more general, and sound events being more detailed. Sound events are components of a wider acoustic scene, allowing a detailed description of what is happening, e.g., “people talking”, “children playing”, “bird singing” possibly being part of a park acoustic scene. In this sense, the acoustic scene considers the sound signal as a whole, while the sound events consider only individual sources. Auditory scene analysis [10] laid out the Gestalt principles for perception of separate sound sources, while experiments in psychology of perception showed that in everyday listening situations humans perceive sound sources rather than physical properties of sounds [11]. The sound sources can therefore be used to label these sound events, for example people, children, birds, and these represent the categories in the machine learning formulation of sound event detection.

There are different theoretical cases for sound event detection, depending on the number of classes involved and the number of labels to be assigned. In general, the term *classification* is used to indicate a multi-class single-label case, where a test audio sample is assigned to a single category, i.e. given a single label. When multiple labels are assigned to the same test audio sample, the task is referred to as *tagging*, while estimation of temporal activity of classes defines the case of *detection* [12]. Specifically, sound event detection involves marking onsets and offsets for multiple instances of sound events within the same test audio sample. All the mentioned cases are topics of the recent series of challenges on Detection and Classification of Acoustic Scenes and Events (DCASE).

The DCASE Challenge has recently received a lot of interest in the research community, and has a continuously growing number of participants. The challenge aims to provide open data for researchers to use in their work, to encourage reproducible research, and attract new researchers into the field. By providing the setting for a competition, and supporting datasets and evaluation tools, the challenge also creates successive reference points for performance comparison. Until now, the challenge has included tasks on acoustic scene classification, sound event

Manuscript received October 26, 2018; revised February 22, 2019; accepted March 14, 2019. Date of publication March 25, 2019; date of current version April 9, 2019. The work of A. Mesaros, T. Heittola, and T. Virtanen was supported by the European Research Council under the ERC Grant Agreement 637422 EVERYSOUND, including the data collection for Tasks 2 and 3. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Tan Lee. *(Corresponding author: Annamaria Mesaros.)*

A. Mesaros, A. Diment, T. Heittola, and T. Virtanen are with the Faculty of Information Technology and Communication Sciences, Tampere University, Tampere 33720, Finland (e-mail: annamaria.mesaros@tuni.fi; aleksandr.diment@tuni.fi; toni.heittola@tuni.fi; tuomas.virtanen@tuni.fi).

B. Elizalde and B. Raj are with the Machine Learning for Signal Processing Group, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: bmartin1@andrew.cmu.edu; bhiksha@cs.cmu.edu).

E. Vincent is with the Universite de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France (e-mail: emmanuel.vincent@inria.fr).

Digital Object Identifier 10.1109/TASLP.2019.2907016

detection in synthetic and real-life audio, and audio tagging [13], [14]. The highest number of participants in all previous editions was in acoustic scene classification, but interest and participation in sound event detection and tagging tasks is continuously growing.

This paper presents the outcomes of the three sound event detection tasks in the DCASE 2017 Challenge[1]. We present these three tasks and discuss the differences in the training approaches imposed by the specific situation: training with highly unbalanced data for detection of rare sound events, supervised and moderately unbalanced data for polyphonic sound event detection in real-life audio, and training using weakly-labeled data. We then report and analyze the results of the methods submitted by the challenge participants, and provide a statistical analysis using confidence intervals, which was not available at the time of publishing the challenge results.

The remainder of this paper is organized as follows. Section II introduces sound event detection in general terms before detailing each task. There is one subsection dedicated to each of the three sound event detection tasks of DCASE 2017, which presents the task definition, the dataset, the experimental setup and the evaluation procedure. The challenge schedule, the submission statistics and the common baseline system for all tasks are presented in Section III. Section IV presents and analyzes the results of submitted systems, including confidence intervals calculation using a jackknife procedure, and Section V analyzes submissions across tasks. Finally, in Sections VI and VII, we present conclusions and future work related to sound event detection tasks within the DCASE Challenge series.

II. Sound Event Detection

Sound event detection is a complex problem, with few different specific situations that differ in interpretation. Detection of the most prominent sound event implies a single label at a time associated with the most prominent sound event, even though there may be other overlapping sounds in the background. As such, *prominent event detection* was the first approach to sound event detection in multisource environments in the CLEAR Evaluation in 2007 [15]. Later, the detection of overlapping events has become of interest, being referred to as *polyphonic sound event detection* in contrast to *monophonic sound event detection* [16] that only provides one label at each time regardless of the overlapping degree; from this point of view, prominent event detection is equivalent to monophonic sound event detection when there are overlapping sounds, because the system detects only one.

Early methods for detection of sound events were adapted from the field of speech recognition, with the use of Gaussian mixture models (GMM) and hidden Markov models (HMM) providing state of-the art results for a few years [17]. Later, nonnegative matrix factorization was employed for overlapping sound event detection, exploiting the additive properties of signal components in complex mixtures [18], [19]. With DCASE 2016, deep neural networks became the method of choice for sound event detection [14]. One clear reason for the choice of deep learning for polyphonic sound event detection is that the structure and training of neural networks directly allow multi-label classification, with multiple neurons in the output layer being trained and allowed to fire at the same time. In comparison, setting up a system based on GMMs or HMMs to provide multiple labels at the same time requires additional effort such as binary per-class setup [20] or multiple Viterbi passes to decode multiple sequences [17].

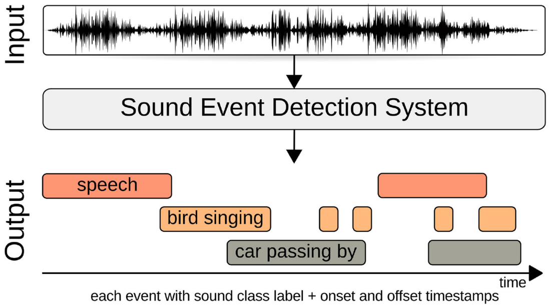

Input
Sound Event Detection System
Output
speech
bird singing
car passing by
time
each event with sound class label + onset and offset timestamps

Fig. 1. Sound event detection in a general case, as required in Tasks 3 and 4.

The sound event detection tasks provided to participants in DCASE 2017 presented three slightly different problems in terms of system training and system output requirements: strongly-labeled training and test data where target sound events are rare, strongly-labeled training and test data with unbalanced classes, and weakly-labeled training data with strongly-labeled test data, as will be explained in the following.

The sound event detection setup familiar from previous DCASE challenges deals with audio material containing target sound events and a reference annotation containing the labels, onset, and offset times of all sound events belonging to the target sound classes. These are referred to as *strong labels*, and usually the training of systems exploits this property for building class models. Target sound events may be overlapping or not, and there may be other overlapping sounds present that are not part of the target classes and are disregarded in the detection setup. An example of system output is illustrated in Fig. 1. This case was the subject of Task 3 and used as training data recordings from everyday environments in which the target sound events have been manually annotated.

Task 2 provided a slightly different setup as detection of rare sound events. In this case, the target sound events were present in about half of the training and test material; the reference annotation contained strong labels and the class models could be trained based on the available instances. Additionally, systems had to cope with large amount of what is considered background audio, which may contain non-target sound events. The task was presented as a binary detection problem, where the target sound class was known. In this case, detection implied finding the onset and offset if the target sound was present in the test audio, similar to the system output illustrated in Fig. 2.

Task 4 presented a third setup for sound event detection. It consisted of sound event detection using weakly labeled training audio from web videos in the context of smart cars; such topics were introduced for the first time in the DCASE series. In

[1] http://dcase.community/challenge2017/

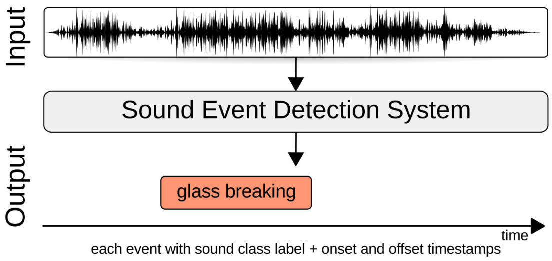

Input
Sound Event Detection System
Output
glass breaking
time
each event with sound class label + onset and offset timestamps

Fig. 2. Sound event detection with rare target sounds, as defined by Task 2.

this setup, the requirement for system output was to produce the onset and offset too, as illustrated in Fig. 1. However, the reference annotation for the training audio had only class labels and no onset and offset information. This means that the sound could be present in any portion of the recording while other multiple unlabeled sounds may be overlapping or occurring in the other portions. Hence, some of the recordings corresponded to more than one label. Moreover, web videos, shared in social media, are typically generated by non-professionals, unstructured, and capturing their everyday lives, hence representing real-world, non-curated audio recordings. The challenge was to train reliable detectors with these recordings and their weak labels. The test part of the development data contained also material having strong labels, to allow benchmarking the methods during the development.

The specific details for each task will be presented in the remainder of this section.

A. Rare Sound Event Detection

The audio material used in Task 2 consists of generated mixtures of background acoustic scenes and target rare sound events. The rareness here stands for the events occurring at most once per 30-second audio instance. Additionally, the event classes were selected to be relevant in potential acoustic surveillance and smart home application scenarios, where they would occur rarely: baby cry, glass break, and gun shot. For each target event class, a separate set of mixtures was provided, and the task was to detect the temporal occurrences of the target events in a single-class detection scenario.

1) Dataset: The dataset consists of the source material for creating mixtures: background scene and target sound event recordings, as well as a set of pre-created mixtures and the software scripts (so-called recipes), according to which they were created. Generation of additional training mixtures with the same source material was permitted, and a software package was provided for such generation, with support for among others arbitrary event-to-background ratios (EBR).

The sound event recordings of the source material were collected from freesound.org using the API and external python tool[2]. There were 209 unique events of baby cries, 197 glass breaks and 263 gun shots in total. The mean duration of the recordings was 2.25 s (baby cry), 1.16 s (glass break), and 1.32 s (gun shot). The material was provided in the original form from freesound.org, accompanied by the strong labels of the temporal occurrences of the events.

Due to the uncontrolled nature of the crowdsourced material, the freesound.org recordings included both target sound events, silence regions and other, non-target sounds. The strong labels of the target sound event occurrences in the original recordings were obtained using a semi-automated procedure, with an automatic segmentation followed by a refinement of the timing and valid sample selection performed by a human annotator. For the details of the procedure, see [9].

For the backgrounds of the generated mixtures of the development and evaluation subsets, the TUT Acoustic Scenes 2016 development and evaluation datasets [20] were used. They include recordings of acoustic scenes of 15 classes such as bus, city center, office, library etc. Prior to mixture generation, manual screening was performed, in which recordings with the naturally occurring sounds similar to the target sound events (mostly, baby cries and shouts) were discarded.

The parameters of the mixture generation were the following. For each target event class in the experimental subsets, 500 signals were created, out of which 250 included target events and 250 were purely the background recordings. The EBRs were selected uniformly randomly from the list of values −6, 0, 6 dB for each mixture. The EBR was calculated in terms of average RMSE values computed over the duration of the event and the corresponding background segment, onto which the event signal would be added. The instances of backgrounds and sound events, as well as the timing of the events in the mixtures were selected randomly and uniformly with a fixed seed of the random generator, thus allowing reproducibility.

The event signals collected from freesound.org had sampling rates of 44.1 kHz and above. Prior to summing, the signals of higher sampling rates were downsampled to 44.1 kHz. The mixtures were saved in 24 bit format in order to minimize quantization noise. The dataset is highly unbalanced due to the temporal rareness of the events within the mixture instances—a problem which might need to be addressed with custom methods.

2) Experimental Setup: The dataset included a development subset, which included training and test partitions and was released at the beginning of the challenge, as well as the evaluation subset, used in the final evaluation and released at a later stage. The split of the generated mixtures into development-training, development-test and evaluation sets was performed in terms of underlying source data. In order to evaluate the systems on truly unseen data, it was crucial to ensure that no sound event or background noise would be shared across the subsets. To further enhance this restriction, the background recordings were split by location ID as provided in the original dataset. The sound events were split by freesound.org user names, so that no recordings from the same user would be present in more than one subset. The meta data indicating the subset to which the source events and backgrounds belong were provided, so that further generated data would be split according to the same rules. The class-wise counts of unique events were the following: baby cry (106 + 42 + 61 instances in training, test and evaluation sets, respectively), glass break (96 + 43 + 58 instances) and gun shot (134 + 53 + 76 instances).

[2] https://github.com/xavierfav/freesound-python-tools

3) Evaluation: The submissions were evaluated for each target event class separately using the event-based error rate (ER) with onset-only condition and a collar of 500 ms [16]. Additionally, the event-based F1-score with the same conditions was calculated but not used in ranking. The systems were then ranked using the average event-based error rate over the three classes.

B. Sound Event Detection in Real-Life Audio

Task 3 of the challenge was the sound event detection setup illustrated in Fig. 1, evaluating performance of sound event detection systems in multisource conditions similar to our everyday life, with the aim of recognizing overlapping events. As in real-life, there is no control over the number of overlapping sound events at each time, nor the number of sound instances present.

1) Dataset: The dataset used for this task is TUT Sound Events 2017, consisting of audio recordings of street scenes with various levels of traffic and human activity. The street environment is of interest for the detection of sound events related to human activities and hazard situations, with wide applicability for personal safety. Audio was recorded in different locations on streets in residential areas and city center, with 3–5 minutes of recording per location, and the recordings were then manually annotated. Individual sound events in each recording were annotated using a noun to describe the source of the sound and a verb to describe the action that produces the sound. The nouns and verbs were chosen from WordNet [21]. The noun-verb pair was used whenever possible, otherwise it was acceptable to use either one of them. The annotator was instructed to annotate all audible sound events, decide the start and end times of the sounds as he sees fit, and choose event labels. A detailed description of the annotation process and more data statistics are provided in [20].

The sound event classes for the TUT Sound Events 2017 dataset were selected based on the resulting annotations, by choosing sounds related to human presence and traffic. The selected sound event classes were: brakes squeaking, car, children, large vehicle, people speaking, and people walking. A mapping was performed between the labels resulting from the initial annotation procedure to these classes, merging sounds into classes described by their source. For example "car passing by", "car engine running", "car idling", etc were all included into the target class "car", sounds produced by buses and trucks were included into the target class "large vehicle", "children yelling" and "children talking" were included into the target class "children", etc. Due to the high level of subjectivity inherent to the annotation process, a verification of the mapped classes in the reference annotation was done as follows; three persons (other than the annotator) listened to each audio segment annotated as belonging to one of these classes, marking agreement about the presence of the indicated sound within the segment. Event instances confirmed by at least one person were kept, resulting in elimination of about 10% of the original event instances.

2) Experimental Setup: The dataset was partitioned randomly into development and evaluation subsets by assigning the available recordings into either set such that the majority of event instances for each class was in the development subset. Event instances for different classes are distributed unevenly within the recordings, therefore the development/evaluation distribution of examples can be controlled only to a certain extent. The resulting number of instances per event class is presented in Table I. Within the development set, a cross-validation setup was also provided for allowing comparison of submissions.

TABLE I
EVENT INSTANCES PER CLASS IN TASK 3 AND AVERAGE LENGTH OF SOUND EVENTS CALCULATED OVER THE ENTIRE DATASET

Event label	Dev. set	Eval. set	Avg. length (s)
brakes squeaking	52	23	2.18
car	304	106	7.64
children	44	15	6.48
large vehicle	61	24	12.89
people speaking	89	37	7.31
people walking	109	42	10.38
total	659	247	

3) Evaluation: Evaluation of submissions was done by calculating the segment-based error rate (ER) and segment-based F1-score with a segment length of one second. In the development set, the metrics are calculated by accumulating error counts (insertions, deletions, substitutions) over all folds before calculating the final values, instead of averaging the individual folds or individual class performance [16]. This method of calculating performance is referred to as *micro-averaging* and gives equal weight to each individual sound instance in each segment, as opposed to being influenced by class balance and error types [22]. Ranking of submitted systems was done by ER, calculated on the evaluation dataset with micro-averaging.

C. Large-Scale Detection of Sound Events Using Weakly Labeled Audio Recordings From Videos

Task 4 evaluated systems for weakly supervised sound event detection in audio recordings from videos in the context of smart cars. The topic of weak labels was chosen due to the abundance and challenge on this type of annotations. This was the first task in DCASE to evaluate audio from videos, which are the main source of recorded sounds. The context of smart cars was chosen due to its industry relevance and the under use of audio. The results of this task helped define new grounds for sound event detection and how it can benefit self-driving cars in smart cities and urban soundscapes. Task 4 consisted of two subtasks, Audio Tagging and Sound Event Detection. We discuss here the latter as illustrated in Fig. 2.

1) Dataset: The task employed a subset of AudioSet [23]. AudioSet consists of an ontology of 587 sound event classes and a collection of 2 million human-labeled 10-second sound clips drawn from YouTube videos. The clips are mono-channel and sampled at 44.1 kHz. The ontology is specified as a hierarchical graph of event categories, covering a wide range of human and animal sounds, musical instruments and genres, and common everyday environmental sounds. To collect the dataset, Google worked with human annotators who listened, analyzed, and verified the sounds they heard within the YouTube 10-second clips. To facilitate faster accumulation of examples for all classes,

TABLE II
EVENT INSTANCES PER CLASS IN TASK 4'S SUBCATEGORIES

Vehicle sounds	Train	Test	Dev	Eval	Avg. length (s)	Warning sounds	Train	Test	Dev	Eval	Avg. length (s)
Truck	6,885	60	6,945	122	5.59	Train horn	345	30	375	108	2.05
Train	2,075	80	2,155	294	5.36	Truck horn	313	30	343	97	2.78
Skateboard	1,516	30	1,546	89	3.38	Car alarm	180	30	210	73	3.95
Motorcycle	3,162	30	3,192	68	7.22	Reversing beeps	245	30	275	63	3.10
Car	25,077	122	25,199	482	5.86	Ambulance siren	524	32	556	62	6.28
Car passing by	3,598	32	3,630	85	3.39	Police car siren	2,364	35	2,399	69	7.79
Bus	3,617	31	3,648	64	6.77	Fire truck siren	2,288	35	2,323	67	6.11
Bicycle	1,897	30	1,927	67	3.09	Civil defense siren	1,409	31	1,440	61	8.57
						Screaming	636	30	666	101	2.54
Total	47,827	415	48,242	1,271		Total	8,304	283	8,587	701	

Development includes training and testing sets. Last column is the average length of the sounds indicated by the strong label.

Google relied on available YouTube metadata and content-based search to nominate candidate video segments that were likely to contain the target sound. Note that AudioSet does not come with precise time boundaries for each sound class within the 10-second clips and thus annotations are considered weak labels. Also, one clip may correspond to more than one sound event class. The numbers of positive labels between classes were imbalanced ranging between 180 for Car Alarm to 25,077 for Car. Task 4 relied on a subset of 17 sound events divided into two categories: *Warning* and *Vehicle*.

1) *Warning sounds:* Train horn, Air horn Truck horn, Car alarm, Reversing beeps, Ambulance (siren), Police car (siren), Fire engine fire truck (siren), Civil defense siren, Screaming.
2) *Vehicle sounds:* Bicycle, Skateboard, Car, Car passing by, Bus, Truck, Motorcycle, Train.

2) Experimental Setup: The data was divided in two main partitions: development and evaluation. The development data was itself divided into training and test. Training had 51,172 clips, which are class-unbalanced and had at least 30 clips per sound event. Test had 488 clips, with at least 30 clips per class. A 10-second clip may have corresponded to more than one sound event class. The evaluation set had 1,103 clips, with at least 60 clips per sound event. The sets had weak labels denoting the presence of a given sound event within the audio, but with no onset and offset annotations. For test and evaluation, strong labels (onset and offset annotations) were provided for the purpose of evaluating performance. The strong labels were collected from the agreement of three human labelers to probe the presence of specific sound event classes in the 10 second clips. The number of instances per class can be seen in Table II.

3) Evaluation: The evaluation metric was segment-based error rate (ER) and F1-score, where ranking of submitted systems was based on ER.

III. DCASE 2017 CHALLENGE

The DCASE 2017 Challenge comprised four tasks. In addition to three tasks outlined above, there was an acoustic scene classification task, and the task dealing with weak labels also had an audio tagging subtask [9]. The phases of the challenge are presented in Table III.

TABLE III
PHASES AND DEADLINES OF THE DCASE 2017 CHALLENGE

Release of development datasets	21 Mar 2017
Release of evaluation datasets	30 June 2017
Challenge submission	31 July 2017
Publication of results	15 Sept 2017
DCASE 2017 Workshop	16-17 Nov 2017

TABLE IV
SUBMISSION STATISTICS FOR ALL FOUR TASKS IN THE DCASE 2017 CHALLENGE

	Teams	Systems	Authors
Task 1	39	96	129
Task 2	13	32	38
Task 3	13	35	32
Task 4	9	55	25

A. Submission Statistics

Each team was allowed to submit the results of a maximum of four systems. The sound event detection tasks received a fair amount of attention, with over 30 systems submitted to each task. Table IV presents the number of participating teams, the total number of submitted systems, and the number of unique authors for each task.

B. Baseline System

The baseline system provided with the data consisted of a common implementation for all tasks. The approach was based on a multilayer perceptron (MLP) [9] and it was provided to serve as a reference point during development, while offering flexibility in building different deep neural network (DNN) architectures on top of it. The audio features used in the baseline system were log mel-band energies calculated in frames of 40 ms with 50% overlap, using 40 mel bands covering the frequency range from 0 to 22,050 Hz. Using a context window of five frames resulted in a feature vector of length 200. The MLP consisted of two dense layers of 50 hidden units each, with 20% dropout, and was trained using the Adam algorithm for gradient-based optimization [24] for maximum 200 epochs using a learning rate of 0.001 and early stopping.

The output layer of the network differs depending on the task. For Task 2 (detection of rare sound events), a separate binary classifier was used for each class, with the output layer consisting of a single neuron with sigmoid activation, indicating the activity of the target class. Detection was done by applying binary classification frame-wise, and integrating the classification decisions into event activity indicator by median filtering with a 0.54 s sliding window with a 20 ms hop. For Tasks 3 and 4, the output layer contained 6 and 17 sigmoid units, respectively, that can be active at the same time, so they could indicate activity of overlapping sound classes. A multi-class multi-label classification was applied frame-wise, and the detection was performed by integrating the classification decision for each class with the same sliding median window of 0.54 s length and with a 20 ms hop.

To obtain the development set performance, the baseline system was trained using the training portion of the development set for each task. Likewise, for each task, the evaluation set performance was obtained by training the baseline system on the full development set and testing it on the evaluation set. For Task 2, the baseline system was trained using the default provided mixtures with no additional data generation. The system performed with ER of 0.53 and an F1-score of 72.7% on the development set, and an ER of 0.64 and an F1-score of 64.1% on the evaluation set. For Task 3, the baseline system had an ER of 0.69 and F1-score of 56.7% on the development set, and an ER of 0.93 and F1-score of 22.8% on the evaluation set. For Task 4, the baseline system had an ER of 1.02 and F1-score of 13.8 % on development set; for the evaluation set the system achieved an ER of 0.93 and an F1-score of 28.4%.

IV. Challenge Results and Analysis

This section presents the challenge results and a detailed analysis of the submissions for the three sound event detection tasks. The complete challenge results, including all the submitted system outputs, have been published to allow further analysis and comparison.[3]

For the statistical analysis of the results, we use confidence intervals for ER and F1-score, calculated using the jackknife estimate. Jackknife approximated confidence intervals result in a rather coarse approximation compared to asymptotic methods; however, asymptotic methods require knowledge of the underlying distribution of the parameter to be estimated, while the jackknife method can be used in cases when the underlying distribution is unknown. By using the jackknife method, we make no assumption on the distribution of our metrics, as for example ER depends on the individual combinations of active sounds at each time within the evaluation segments.

The jackknife is a resampling technique used for estimating a parameter from a random sample of data for a population, based on partial estimates. Using these partial estimates, a bias-corrected jackknife estimate of the parameter of interest can be calculated, as well as its variance and confidence intervals [25]. Calculation starts with estimating the parameter from the whole sample, in our case the ER and the F1-score being measured from all the evaluation data. Then the partial estimates are calculated with a leave-one-out method, leaving out in turns one observation in the calculation—in our case excluding in turn each file or 1 s segment from the calculation of the ER and the F1-score, depending on the evaluation method. Pseudo-values are calculated as the difference between the whole sample estimate and the partial estimates, and these pseudo-values are further used for calculating the jackknife estimate of the parameter, which is bias-corrected, and its standard deviation. In this procedure, the observations are assumed to be independent of each other, i.e. independent and identically distributed (i.i.d.).

For a given quantity θ, the sample estimate $\hat{\theta}$ based on a sample of N observations is a function of the observations:

$$\hat{\theta} = f(X_1, X_2, \ldots, X_N). \tag{1}$$

The estimators obtained by leaving out sample X_i are:

$$\hat{\theta}_{(i)} = f(X_1, X_2, \ldots, X_{i-1}, X_{i+1}, \ldots, X_N) \tag{2}$$

and

$$\hat{\theta}_{(.)} = \frac{1}{N} \sum_i \hat{\theta}_{(i)}. \tag{3}$$

The n pseudo-values are:

$$\tilde{\theta}_i = N\hat{\theta} - (N-1)\hat{\theta}_{(i)}. \tag{4}$$

The jacknife estimate $\hat{\theta}_{\text{jack}}$ is obtained as the mean of the pseudo-values:

$$\hat{\theta}_{\text{jack}} = \frac{1}{N} \sum_i \tilde{\theta}_{(i)}. \tag{5}$$

With the pseudo-values considered to be independent random variables, the standard error of the parameter estimate can be obtained from the variance of the pseudo-values as the standard error of the mean:

$$\hat{\sigma}_{\tilde{\theta}_{\text{jack}}} = \sqrt{\frac{\hat{\sigma}^2_{\tilde{\theta}}}{N}} = \sqrt{\frac{\sum_i (\hat{\theta}_{(i)} - \hat{\theta}_{(.)})^2}{N(N-1)}}. \tag{6}$$

For i.i.d. observations, this estimation follows a Student's t distribution with $(N-1)$ degrees of freedom, and the $(1-\alpha)$ confidence interval can be computed as:

$$\hat{\theta}_{\text{jack}} \pm t_{\alpha,\nu} \hat{\sigma}_{\tilde{\theta}_{\text{jack}}} \tag{7}$$

For each task, we used this method to compute the jackknife estimate and 95% confidence intervals for the ER and the F1-score as used for evaluation (event-based for the rare sound events detection, segment-based for the others).

A. *Task 2 - Rare Sound Events Detection*

A total of 13 teams participated in Task 2 producing 32 submitted systems. The statistical analysis (see the F1-score and ER estimates and confidence intervals in Fig. 3 and Table V) was performed for the top system of each team. The two top teams (Lim [26] and Çakır [27]) submitted systems whose error rate estimates are within the 95% confidence interval of each other,

[3] https://doi.org/10.5281/zenodo.2598364

TABLE V
SUMMARY OF SYSTEMS SUBMITTED FOR RARE SOUND EVENT DETECTION (TASK 2), BEST PER TEAM

System	Features	Classifier	F1 (%)	$\mathrm{F1}_{\mathrm{JK}} \pm 95\%$ CI (%)	ER	$\mathrm{ER}_{\mathrm{JK}} \pm 95\%$ CI
Lim_COCAI_1	log-mel energies	CRNN	93.14	93.15 ± 1.42	**0.1307**	0.1306 ± 0.0261
Çakır_TUT_2	log-mel energies	CRNN	90.97	90.98 ± 1.74	**0.1733**	0.1732 ± 0.0326
Phan_UniLuebeck_1	log Gammatone cepstral coef.	DNN+CNN	85.34	85.36 ± 2.15	**0.2773**	0.2771 ± 0.0394
Zhou_XJTU_1	spectrogram	NMF	84.16	84.18 ± 2.12	**0.3133**	0.3129 ± 0.0422
Liping_CQU_3	spectrogram	CNN	82.00	82.02 ± 2.36	**0.3173**	0.3172 ± 0.0400
Vesperini_UNIVPM_1	log-mel energies	MLP	83.92	83.94 ± 2.28	**0.3267**	0.3262 ± 0.0471
Dang_NCU_2	log-mel energies	CRNN	79.07	79.09 ± 2.48	**0.4107**	0.4102 ± 0.0515
Ravichandran_BOSCH_4	log-mel spectrograms, MFCC	MLP, CNN, RNN	78.55	78.57 ± 2.62	**0.4267**	0.4263 ± 0.0525
Wang_BUPT_1	log-mel energies	DNN	73.40	73.44 ± 2.86	**0.4320**	0.4318 ± 0.0422
Wang_THU_1	MFCC, log-mel energies	DNN, HMM	72.57	72.60 ± 2.91	**0.4973**	0.4970 ± 0.0503
Li_SCUT_4	MFCC	FC+Bi-LSTM	69.80	69.81 ± 2.80	**0.6000**	0.5997 ± 0.0628
Baseline	log-mel energies	MLP	64.12	64.14 ± 3.12	**0.6373**	0.6371 ± 0.0588
Jeon_GIST_1	log-mel energies from NMF	MLP	65.78	65.77 ± 3.27	**0.6773**	0.6768 ± 0.0759

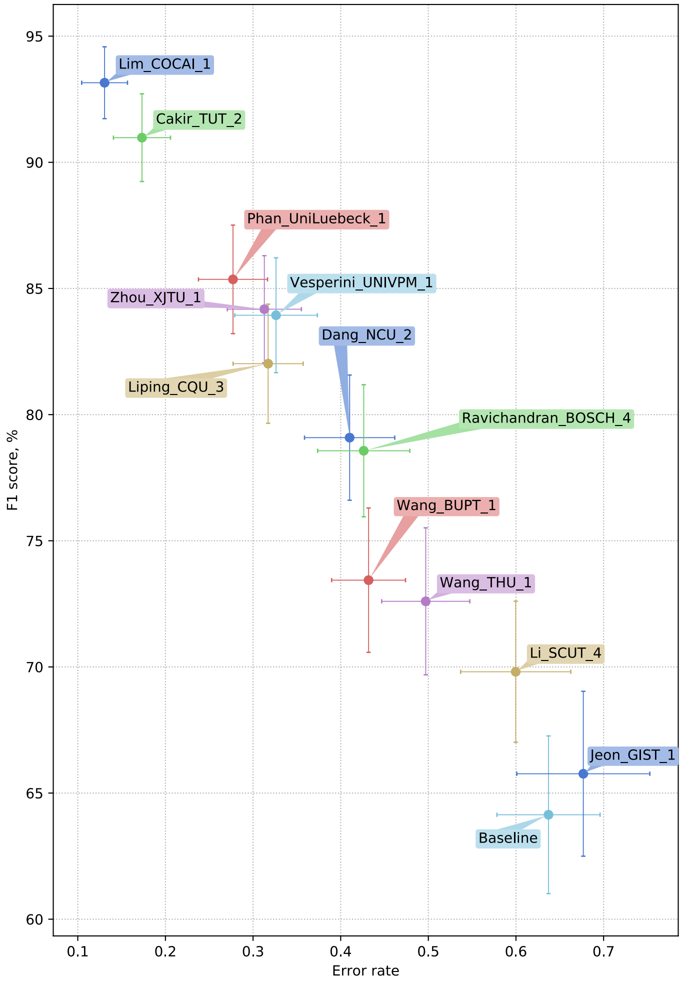


Fig. 3. Scatter plot of submitted systems for Task 2 accompanied by the 95% confidence intervals. Systems were ranked by ER (the smaller the better).

while being clearly non-overlapping from the rest of the submissions. This suggests more confidently that these two systems are indeed best among all submissions. The overall relatively moderate range of the confidence intervals of all the analyzed systems indicate that the size and diversity of the provided dataset was adequate for the task.

Out of the 32 submitted systems, 13 generated additional mixtures using the provided source data and mixture generation software. The authors of the winning system (Lim *et al.* [26]) performed this most extensively, generating four different training sets of 5000 mixtures each (4 × 10 times the size of the original dataset). Vesperini *et al.* [28] performed additional mixture generation only for the class "gun shot", motivated by the relative short duration of the sounds of this class. To address this, they created 500 additional gun shot mixtures, yielding the full development set 1.2 larger than the original one. Çakır *et al.* [27] generated a training dataset 2.2 times larger than the original one, while addressing the class imbalance problem by adjusting the parameters of the mixture generation. They increased the event occurrence rate from 0.5 to 0.99, producing almost all the mixtures in the training set with the event present and thus doubling its frame-wise occurrence.

In addition to the aforementioned technique of generating mixtures with higher target event occurrence rates, other methods for addressing the imbalance issue were applied. Phan *et al.* [29] performed resampling and used a weighted loss, penalizing false negatives ten times more than false positives. Wang and Li [30] tackled the imbalance problem with a dynamic decision threshold computed from the average system output value. Finally, Wang *et al.* [31] used differently weighted update rules based on the event presence in each frame.

Log-mel energies were the most used features (20 systems), followed by mel frequency cepstral coefficients (MFCCs, in 9 systems). The frame-blocking parameters were similar across the systems, with the frame length values of 40 ms with 50% overlap being used by 10 systems. Most of the systems (19) used stacking of consecutive frames, with the context size of 5 frames being the most popular, while some participants optimized this hyperparameter extensively. The winning system by Lim *et al.* [26]) used a different kind of context processing by segmenting the feature sequences in a sliding manner into short "macroblocks" of 5 to 100 frames (class-specific hyperparameter), performing the inference on those signals and averaging the outputs over multiple overlapping excerpts. Almost all the classifiers were neural network based, with most popular architectures being CRNNs (12) and CNNs (9). Ten systems used ensemble technique to obtain final outputs.

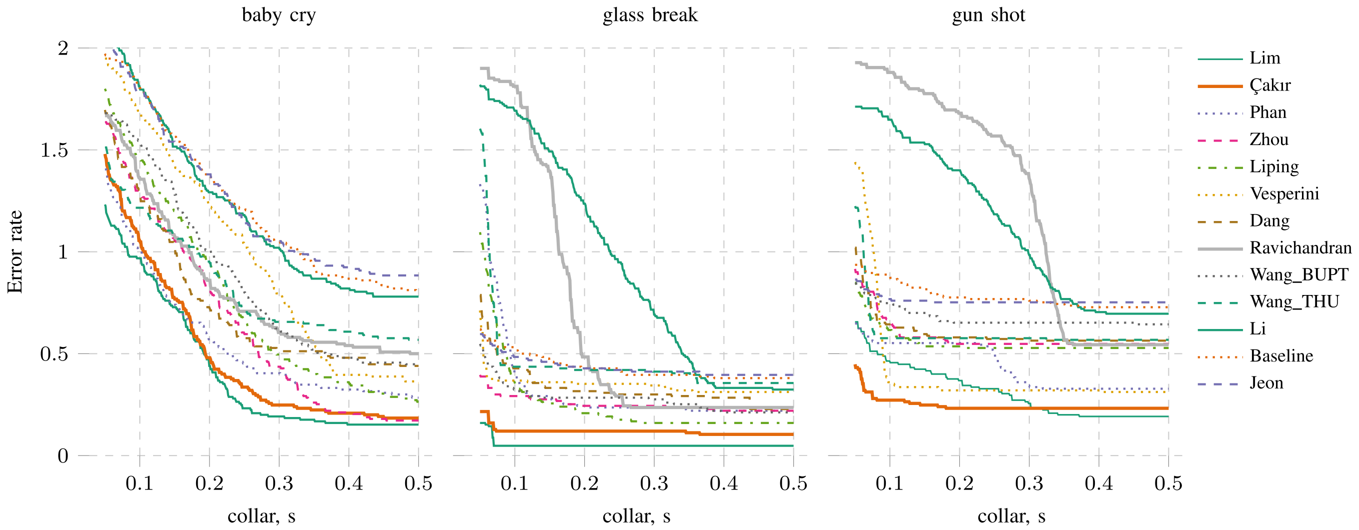


Fig. 4. Class-wise event error rate as a function of a time collar — tolerance used when evaluating validity of the onsets (best system per team). In the challenge ranking, the collar value of 0.5 s was used, and the systems are ordered in the legend accordingly. Solid lines emphasise the systems whose behaviour differs from the majority and is addressed in the text.

Systems in the top ranks managed to generalize well to the unseen evaluation data. Out of the top ten systems, nine had less than 0.1 difference between the error rates on the development-test and the evaluation sets. Three systems out of top ten had error rate differences of less than 0.05.

The submitted systems showed a consistent difference in class-wise performance, with glass break being the easiest class in terms of F1-score and error rate for all the submitted systems. The average class-wise F1-scores over top ten systems were 90.02% for baby cries, 93.82% for glass breaks and 85.42% for gun shots.

Additional insight can be gained by analyzing the performance of the systems with a varying degree of temporal tolerance of onset detection — the so-called time collar. In the challenge ranking, a value of 0.5 s was used as the maximum allowed temporal mismatch between the predicted and ground truth onsets for the events to be deemed detected correctly. In Fig. 4, we perform the evaluation with a varying collar value. For the baby cry class, we see that the systems show similar sensitivity to the value of time collar. For the other two classes, we observe certain peculiarities.

With the glass break class, most of the systems reach their peak performance already at very small values of time collar (70-160 ms), while in the cases of systems Ravichandran_BOSCH_4 and Li_SCUT_4, the error rate saturates much later (250 and 400 ms, respectively). This indicates that in most cases when the systems are capable of classifying the highly impulsive signal correctly, their onset detection is extremely accurate.

With the gun shot signals, which are also impulsive, but might include longer reverberation tail, certain systems show similar properties. For instance, the error rate of Çakır_TUT_2 saturates already at collar values of 110–180 ms and is the best across all the systems in the low-collar range. The overall winning system Lim_COCAI_1, however, appears to improve gradually with increasing collar value, which can be explained by the fact that it uses ensembling over much larger number of consecutive macroblocks (of sizes 10, 14, 20 and 50 frames) than in the case of glass breaks (macroblocks of 5 frames only). If accurate onset detection of impulsive signals is preferred, Çakır's approach of incorporating neighbouring frame information using a simple median filtering of the frame-wise predictions is more successful than class-specific hyperparameter-optimized ensembling of networks operating with different time resolutions.

Similarly to the glass break class, systems Ravichandran_BOSCH_4 and Li_SCUT_4 are outliers in terms of their performance over different time collars with the gun shot class as well. Li_SCUT_4 improves mostly linearly. This can be attributed to the system being developed jointly for Tasks 1-3 and not extensively optimized for the current task or class. Additionally, it is one of the few systems to use MFCC features fed into a fully-connected network as opposed to the predominant log-mel energies fed into a CNN. Ravichandran_BOSCH_4 shows an interesting rapid improvement at around 310 ms. This can be explained by the unusually large frame lengths and context sizes used by this system: it consists of an ensemble of an RNN fed with frames of 120 ms length and 50% overlap and a CNN operating on 16-frame blocks (frame length 100 ms, overlap 80%).

This analysis shows that optimizing the systems for the task and event classes is reasonable when the events are of different nature (harmonic vs. percussive) while complex ensembling and overly extensive optimization leads to good results at the expense of robustness to the parameters of the evaluation metric.

B. *Task 3 - Sound Event Detection in Real-Life Audio*

For Task 3 there were 35 submitted systems, originating from 12 different teams. Table VI presents a summary that includes only the best system of each team. The visualization of the systems' ranking, including 95% confidence intervals, is presented in Fig. 5.

TABLE VI
SUMMARY OF SYSTEMS SUBMITTED FOR SOUND EVENT DETECTION IN REAL-LIFE AUDIO TASK (TASK 3)

System	Features	Classifier	F1 (%)	$F1_{JK} \pm 95\%$ CI (%)	ER	$ER_{JK} \pm 95\%$ CI
Adavanne_TUT_1	log-mel energies	CRNN	41.7	41.7 ± 2.3	**0.7914**	0.7914 ± 0.0268
Lee_SNU_3	log-mel energies	CNN	40.8	40.8 ± 2.2	**0.8080**	0.8079 ± 0.0275
Lu_THU_1	MFCC, pitch	RNN, ensemble	39.6	39.6 ± 2.3	**0.8251**	0.8250 ± 0.0294
Zhou_PKU_1	log-mel energies	LSTM	39.1	39.1 ± 2.2	**0.8526**	0.8525 ± 0.0310
Chen_UR_1	log-mel energies	CNN	30.9	30.9 ± 2.5	**0.8575**	0.8575 ± 0.0231
Xia_UWA_3	log-mel energies	CNN	41.7	41.7 ± 2.1	**0.8740**	0.8738 ± 0.0340
Kroos_CVSSP_2	scattering transf, clustering	Neuroevolution	41.6	41.6 ± 2.2	**0.8911**	0.8909 ± 0.0355
Hou_BUPT_2	raw audio	BGRU	34.1	34.1 ± 2.3	**0.9248**	0.9246 ± 0.0334
Baseline	log-mel energies	MLP	42.8	42.8 ± 2.0	**0.9358**	0.9355 ± 0.0404
Dang_NCU_2	log-mel energies	CRNN	42.8	42.8 ± 2.1	**0.9468**	0.9465 ± 0.0429
Li_SCUT_2	DNN(MFCC)	Bi-LSTM	41.0	41.0 ± 2.0	**0.9523**	0.9520 ± 0.0415
Wang_NTHU_1	MFCC, TDOA	RNN	40.8	40.8 ± 2.1	**0.9749**	0.9746 ± 0.0458
Feroze_IST_2	PLP	NN	39.7	39.7 ± 2.1	**1.0312**	1.0309 ± 0.0489

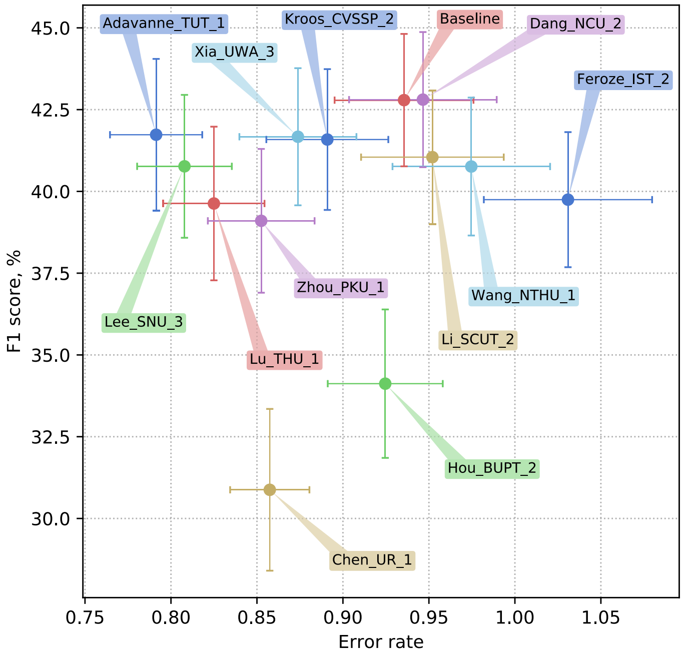


Fig. 5. Scatter plot of systems in Table VI. Systems were ranked by ER.

The systems were evaluated and ranked using the segment-based error rate (ER) in one-second segments.[4] Overall, there were 19 systems that outperformed the provided baseline performance in terms of ER, which was 0.93 on the evaluation dataset. Several of these top systems were submitted by the same team: in terms of teams, eight different teams outperformed better baseline on the evaluation dataset, with the top performance being 0.79. The F1-score was also calculated in one-second segments to provide a more complete characterization of the systems' performance. In terms of F1-score, the baseline system had the best performance on the evaluation set, at 42.8%, together with one submitted system. The top-ranked system by Adavanne *et al.* [32] (with smallest ER) had the second-highest F1-score of 41.7%.

Confidence intervals are presented in Table VI and Fig. 5 for the top system of each team. For this task, the leave-one-out partial estimates were calculated in accordance with the evaluation process, by considering each one-second segment as a separate sample. The independence assumption for these segments is used implicitly in the metrics, by determining the number of true positives, deletions, insertions and substitutions separately for each segment before the final metric calculation [16]. Based on the confidence intervals, we can see that the performance of consecutively ranked systems is not significantly different, with both the ER and the F1-score degrading gracefully with the top four systems. In terms of F1-score, the overlaps are much more prominent, with 10 of the 12 considered systems being within 4% average performance of each other and also very close to the baseline system. The top four systems are, however, performing significantly better than the baseline in terms of ER, while there is also one system performing significantly worse. At the same time, the submission by Chen *et al.* [33], ranked fifth, while having significantly better performance than the baseline in ER, has significantly lower in F1-score than other systems.

There is not very much diversity in the characteristics of the submitted systems, with many submissions using similar features and classifiers among other processing steps. Among all 36 systems, 21 used the single-channel audio input as provided in the baseline system (averaging the two channels), while 8 systems used the binaural audio and few others used audio in mixed ways (difference, mean, right channel only). Most of these cases used the two channels and their combinations as a method for data augmentation, given that the audio recorded by the two microphones is slightly different. Only two teams used established data augmentation methods such as channel swapping (4 systems of one team) and pitch shifting and time stretching (4 systems of one team).

The most popular features were mel-based representations, with 19 systems (including [34], [35]) using log mel energies, including multiscale versions, and 9 systems using mel-frequency cepstral coefficients (MFCC); in some cases, MFCCs were used in combination with other features such as pitch [36], or TDOA. One system used raw audio as input, while three systems

[4] This discussion excludes the system by Yu. The submitted system had very low scores, attributed to a software bug. The system was resubmitted after the deadline and yielded substantially lower overall ER, but is not included in the official ranking.

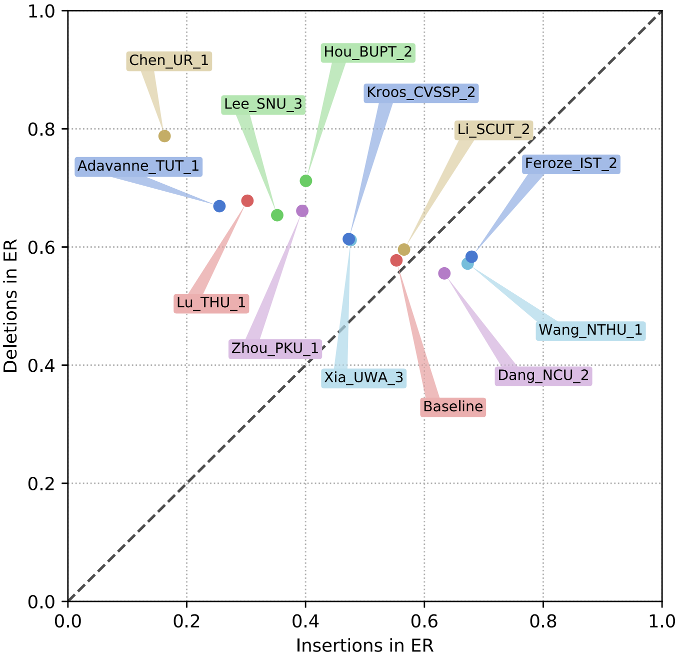
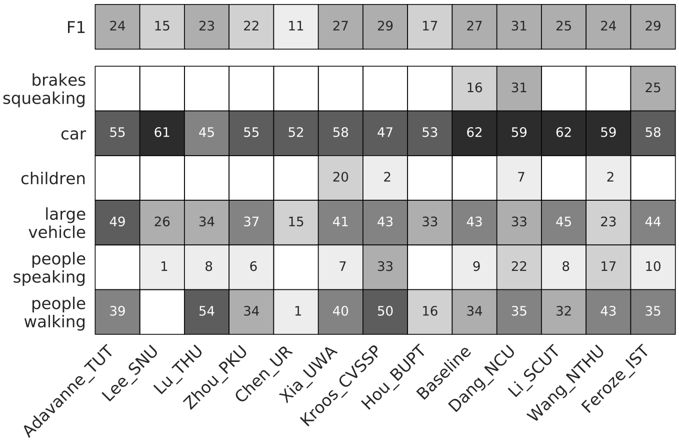


Fig. 6. Deletions and insertions contribution to ER for systems in Table VI.



	Adavanne_TUT	Lee_SNU	Lu_THU	Zhou_PKU	Chen_UR	Xia_UWA	Kroos_CVSSP	Hou_BUPT	Baseline	Dang_NCU	Li_SCUT	Wang_NTHU	Feroze_IST
F1	24	15	23	22	11	27	29	17	27	31	25	24	29
brakes squeaking									16	31			25
car	55	61	45	55	52	58	47	53	62	59	62	59	58
children						20	2			7		2	
large vehicle	49	26	34	37	15	41	43	33	43	33	45	23	44
people speaking		1	8	6		7	33		9	22	8	17	10
people walking	39		54	34	1	40	50	16	34	35	32	43	35

Fig. 7. Class-wise F1-score of top system per team, according to Table VI. Empty cells represent classes that were not detected by each system.

belonging to the same team used scattering transform and clustering as feature representation.

In terms of classifiers, deep learning based methods were the most popular, with 24 systems being based on various architectures such as CNN (7 systems), CRNN (8 systems), RNN (5 systems), including LSTM (2 systems) and BLSTM (3 systems). There was not so much use of ensemble classifiers as in the previous DCASE sound event detection task, nor was there any use of classical pattern recognition methods such as GMMs, SVMs and such. Neuroevolution stands out within this challenge as a novel method [37], combining genetic algorithms with an artificial neural network for evolving the weights and topology of neural networks, with the aim of obtaining a small-size model. The evolved small networks performed relatively well on the development data, but significantly worse on the evaluation set.

As we see in this overview, there are not many differences between systems' characteristics; it often seems to be a matter of finding a good operation point for achieving the desired outcome, and optimization of the system following a criteria linked to the evaluation metric. We analyze more closely the contribution of insertions and deletions to the error rate for the systems in Table VI, illustrated in Fig. 6.

We observe that all leading systems have much higher deletion than insertion rates (w.r.t. number of reference events, as part of ER). Due to the specifics of its calculation, minimizing ER can be achieved first of all by making as few errors as possible, secondly by making about the same number of insertions as deletions (therefore counting them as substitution errors). From Fig. 6, we observe that most systems are situated on one side of the diagonal, which means that according to the optimization functions used for training them, it is optimal to output a small amount of positives (producing deletions, but output mostly correct events) rather than a high amount of positives (producing insertions, but also correct events). This behavior is also reflected in other metrics, with systems having generally a high precision (they output mostly correct positives) but a small recall (they miss many of the positives).

Class-wise detection results reveal large differences between the target classes, as illustrated in Fig. 7 by the F1-scores (contains only systems from Table I). The figure reveals that "car" class was easiest to detect, while "brakes squeaking" and "children" were rather difficult. For many systems, some classes go completely undetected (e.g. "children" not detected by 25 of 36 systems).[5] Another notable aspect is the systems' behavior with regard to the class-wise balance of detection performance, observable in the columns of Fig. 7. Interestingly enough, top systems only detect 3 or 2 of the 6 classes, which is reflected in the macro-average F1-score (average of class-wise F1-scores, as opposed to micro-average where the number of true positives, etc. is accumulated across all classes [16]). On the other hand, the system by Kroos (neuroevolution) [37] has a macro F1-score of 29%, compared to Adavanne *et al.* [38] at 23%, while the most balanced output class-wise is a variant of the Dang_NCU submission (not the one in Fig. 7), having the highest macro F1-score among all submissions (37%). Unfortunately, it also has a very high false positive rate that ranks it very low among submissions (rank 30).

The top ranked system by Adavanne *et al.* [32] with an ER of 0.79 achieves an overall F1-score of 41.7%. The difference between its performance on the development set and evaluation set is the extremely large ER of 0.25 vs. 0.79 and F1-score of 79.3% vs 41.7%–possibly indicating overfitting. The system actually does not detect at all one of the six target classes (brakes squeaking), and detects only few instances of children and people speaking, but erroneously ($F1 = 0$, $ER > 1$).

An important observation based on this analysis is the preference of deep learning methods to output a small amount of positives, with this phenomenon affecting mostly the classes for

[5] Detailed class-wise ER and F1-scores for all systems can be accessed on the challenge website, http://dcase.community/challenge2017/task-sound-event-detection-in-real-life-audio-results.

TABLE VII
SUMMARY OF THE SYSTEMS SUBMITTED FOR LARGE-SCALE DETECTION OF SOUND EVENTS USING WEAKLY LABELED AUDIO RECORDINGS (TASK 4)

System	Features	Classifier	F1 (%)	$F1_{JK} \pm 95\%$ CI (%)	ER	$ER_{JK} \pm 95\%$ CI
Lee_SNU_4	log-mel energies	CNN, ensemble	55.5	53.92 ± 3.19	**0.66**	0.6720 ± 0.0319
Xu_CVSSP_1	log-mel energies	CRNN	51.8	51.78 ± 3.09	**0.73**	0.7337 ± 0.0309
Lee_KAIST_4	raw waveforms	CNN	47.1	47.10 ± 3.06	**0.75**	0.7527 ± 0.0306
Salamon_NYU_4	log-mel energies	ensemble	45.9	45.87 ± 3.21	**0.77**	0.7652 ± 0.0321
Adavanne_TUT_4	log-mel energies	CRNN	49.0	49.04 ± 4.07	**0.79**	0.7863 ± 0.040
Chou_SINICA_1	spectrogram	CNN	42.4	42.45 ± 3.02	**0.83**	0.8326 ± 0.0302
Toan_NCU_4	log-mel energies	DenseNet	41.6	41.59 ± 2.70	**0.87**	0.8680 ± 0.0270
Baseline	log-mel energies	MLP	28.4	28.38 ± 2.94	**0.93**	0.9264 ± 0.0294

which there were less examples in training. Their optimization with respect to ER as a metric therefore compensates for the inability of learning the small classes by learning well the larger ones, and producing a mostly correct output. As a general conclusion, for this task and with this benchmark dataset, we cannot state that any deep learning architecture offers superior performance compared to others, therefore the field is still open for investigation.

C. Task 4 - Large-Scale Detection of Sound Events Using Weakly Labeled Audio Recordings From Videos

Task 4 had 55 submissions from 9 different teams as shown in Table IV, among which 31 corresponded to Subtask A: Audio Tagging and 24 to Subtask B: Sound Event Detection. A summary of Subtask B systems and their performance is provided in Table VII. A visualization of the performance and ranking is shown in Fig. 8. The systems were evaluated and ranked using the segment-based ER in one second segments. All the submissions outperformed the baseline performance on the evaluation set in terms of ER, which was 0.93, with a top achieved ER of 0.66 [39]. The F1-score was also calculated for each 10 seconds clip to understand better the performance of the systems. All the submissions outperformed the F1-score baseline of 28.4%, and the top-ranked system achieved 55.5% [39].

Confidence intervals are presented in Table VII and Fig. 8 to statistically compare the systems' performance. The confidence intervals show that the systems performed significantly better than the baseline in terms of ER and F1-score. The system ranked first is better than the rest, but there is a cascaded overlap between the second and the seventh place.

The overall performance of *Warning sounds* was better than *Vehicle sounds* across all systems. The results are consistent with what we expect from human hearing. Warning sounds (i.e. sirens, honks and alarms) are explicitly designed to be heard by humans due to their properties, such as fast increase in amplitude, high loudness and high frequencies. The classes with highest and lowest F1-score (and ER) are the following, *Warning*: Civil defense siren 85.8% (0.29) and Ambulance siren 25.7% (0.9), where 11 systems achieved less than 1% (1.0). We concluded that although there were 4 types of sirens, they were typically detected as Civil defense siren, which after inspection, seemed to be the most distinctive out of all; *Vehicle*: Train 75.4% (0.51) and Car passing by, which was not detected by 15 systems and was mostly labeled as Car. It is hard to tell why Train performed better than other classes like Motorcycle. After inspection, we found that in general, Train recordings did not co-occur with other vehicles, in contrast to Bus, Truck and Car. This explanation also applied to Skateboard.

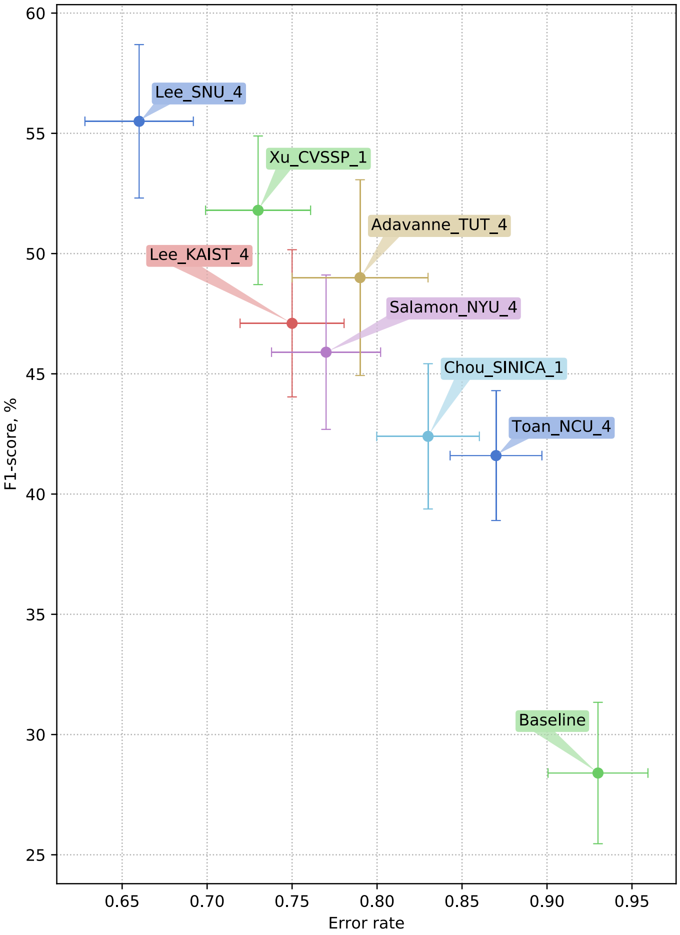


Fig. 8. Scatter plot of the submitted systems for Task 4 Subtask B. The systems were ranked by ER (lower is better). The best performing system in terms of both ER and F1-score is the one closest to upper left corner.

Classes had an imbalanced number of samples as shown in Table II, but this was not strongly related to performance. For example, Car and Truck with 25,077 and 6,885 training samples had a corresponding best detection of 67% (0.73) and 46.9% (0.92) and ranked in seventh and fifteenth out of the 17 classes based on ER. Conversely, Car alarm and Reversing beeps had 180 and 245 training samples with a corresponding best detection of 58.6% (0.66) and 52.4% (0.74) and ranked in fourth and eighth. Civil defense siren was first with 1,409 training samples and 85.8% (0.29). Car passing by was last with 3,630 training samples and 24.1% (1.0). It is to be noted that the development set includes at least and about 30 samples per class for testing and the rest are for training. Participants typically used the entire development set to train their systems for evaluation. In general, there was no overfitted system to the development set and systems performed better in the evaluation set.

About 9% of the recordings had two or three labels which implied overlapping sounds within the recordings. The labels are derived from an ontology defined by the authors of AudioSet. For example, sound classes that could naturally co-occur together, such as Truck and Truck Horn. This is a common problem in real-life audio where sounds rarely occur in isolation and often overlap. Moreover, we included the class Car, which although sometimes was the super class of, for example, Police car siren and Car passing by, it mainly occur independently. After inspecting the class Car, it evidenced a wide acoustic diversity, such as the sound of the car engine but with the car still, the sound of the car moving recorded from the inside, and from the outside, and the sound of the car changing gears. The case of multi labels corresponding to super classes poses an interesting reflection on how should we define sound ontologies, labels and how should we use them for training sound event detectors.

The challenge of training systems using weak labels was approached differently. The best performances came from systems that processed the weak labels. Table II shows, for the evaluation set, the average duration of a sound event based on the strong labels. Note that the average duration across classes is 5 seconds, which is 50% of the provided 10-second clips, hence, the importance of exploring the impact of weak labels in this task. Lee *et al.* [39] used a global-input and separated-input system to process an audio recording as a whole and per segment. Then, the output was used to perform weak (whole recording) and strong (per segment) predictions. Xu *et al.* [40] used a neural network architecture with attention, which ideally focused on regions where the target sound occurred. Salomon *et al.* [41] used multiple instance learning, where training data are labeled as bags of examples, a bag is positive if any of its constituent examples are positive, and negative if none of its examples are positive. Adavanne *et al.* [38] did not pre-process the weak labels. However, they used an architecture that branches out into both, weak and strong label outputs, hence learning from both types of annotations and performing both types of predictions. Lastly, there were two participants which treated weak labels as strong labels.

All the systems were based on log-mel spectrogram features and convolutional neural networks. Two participants preprocessed the raw audio. Lee *et al.* [39] normalized the amplitude of each audio signal using its corresponding absolute maximum amplitude value and then applied background noise removal by subtracting the median value. Salamon *et al.* [41] used data augmentation in the form of pitch shifting and dynamic range compression. The log-mel spectrogram features were extracted with different window sizes across systems ranging from 1 to 5 seconds, with typically 50% overlap and with a range of 40 to 128 mel filterbanks. Lee *et al.* [42] employed raw audio signal as the input of their system, which internally used a multi-level and multi-scale CNN to extract spectrograms with optimal parameter values, such as hop size and window size. The features were then passed to CNNs, which had a number of convolutional layers that varied from 3 to 16 and the number of filters were up to 128. Three teams [38], [41], [43] considered temporal information by adding recursive neural networks to their architectures. Interestingly, the top six submissions from three different participants did not contemplate the temporal information.

Further post-challenge research was carried by some other authors. For example in [44], the authors used an audio-visual approach to match co-occurrences of images and sounds to locate the target sound from the weak labels. The authors in [45] used a multi-level attention model to focus on the target sound indicated by the weakly labels. Lastly, inspired by DCASE 2017 Task 4, a similar task was organized in DCASE 2018 called Large-Scale Weakly Labeled Semi-Supervised Sound Event Detection in Domestic Environments [46].

V. Cross-Task Submissions

A few teams submitted to multiple tasks. For example Dang *et al.* [43] submitted systems using the same architecture to all four tasks (systems Dang_NCU in Tasks 2 and 3, Toan_NCU in Task 4). The systems used parallel networks, one CNN and one RNN, with the purpose of learning spatial information of multidimensional data with the CNN, and learning temporal sequential information with the RNN. Slightly different features and temporal resolutions were used for the different classes and tasks: 40 log mel energies calculated in 40 ms window size and 20 ms hop for Baby crying and Glass breaking classes, 20 MFCCs with delta and acceleration (60-dimensional) for the gunshot class in Task 2; 20 ms window size with 10 ms hop for all classes and the same 60-dimensional vector of MFCC coefficients in Task 3; 64 log mel energies calculated with window size of 1024 and hop size of 256 ($\approx$46 ms and $\approx$12 ms for audio downsampled to 22050 Hz) in Task 4. Regarding system optimization, they mentioned within the Task 4 description that the optimal set of parameters is determined at the peak of F-score of the development set—which explains their ranking in the lower half of the board; the systems being optimized using F1-score, ranked low in terms of ER.

Another system architecture submitted to multiple tasks (Tasks 1, 2 and 3) was submitted as Li_SCUT [47], and was based on extracting bottleneck features (Deep Audio Features) using frame-based processing, and Bi-LSTM for classification. The authors provide no further details about parameter optimization per task, only mention that the classification output is different depending on the task (single or multi label). The systems ranked in the bottom quarter in Tasks 2 and 3.

Adavanne_TUT systems [38] used the same system architecture for Tasks 3 and 4, with the main difference in the number of nodes in layers to accommodate the larger amount of data in Task 4. The method did not treat the strong labels in Task 3 and the weak labels in Task 4 any differently: for training with weak labels, the system simply considered the annotated target sound as being active throughout the entire training sample, a procedure recently called *strong label assumption* [48] or *false strong labeling* [49]. The method ranked first in Task 3, but only 7th in Task 4, which indicates that strong labels have a significant impact on model learning; however, in Task 4, the system generalized much better than in Task 3 (see Sec. IV-B), having similar performance on development and evaluation datasets, likely as a result of sufficient amount of development data to avoid overfitting.

Lee_SNU systems [35], [39] took a completely different approach, submitting different systems to tasks 3 and 4. For Task 3, they used a CNN with inputs of different time length, short-term data, corresponding to ≈4 s, and long-term data, corresponding to the entire audio file (3-5 minutes), for which log-mel and long-term averaged log-mel are merged within the network. The system was specifically designed to detect events with 1 s time resolution, corresponding to the evaluation metric segment length. One interesting detail is that their best system, ranked second in Task 3, was an ensemble of 3 models, while the other ones were ensembles of 4 models, with one model calculated in each fold. The authors mention they worried that the exceedingly poor performance in fold 1 might mean that the fold 1 model failed learning.

For Task 4, the Lee_SNU systems consist of an ensemble of CNNs that use various analysis windows, having a global-input model that uses the entire length of the clip, and multiple separated-input models corresponding to segments of 1, 2, 3, 4, and 5-seconds from the audio clip, with a 1-second sliding window. The models were optimized using F1 or ER, corresponding to each subtask, and ensemble selection was performed by repeating iterations and adding a model that maximizes performance at each point. All 4 systems ranked top (one was ranked first, 3 ranked second, with same performance), and had significantly better performance than the competitors.

These systems, along with the baseline system, show that indeed different sound event classification and detection tasks can be solved using the same core method. In some cases, such approach may provide satisfactory results. However, careful consideration of the problem and task-specific design will likely produce better performance. When drawing inspiration from the solution to other problems, one must not forget to adapt the method to the problem at hand, with more than hyper-parameter optimization.

VI. Conclusions

The DCASE Challenge has already become a familiar yearly event, and continues to develop under the influence of popular research directions and the spur of open datasets creation and publication that it has generated.

One lesson learned from DCASE 2017 (and, implicitly, DCASE 2016), is that sound event detection in real-life audio based on strong labels is not a suitable task for the time being. This is largely due to the scarcity of available strongly-labeled data, and this problem is not likely to be solved soon due to the difficulty of obtaining annotations for a sufficiently large dataset. The currently available data amounts that are strongly labeled are limiting in the size and depth of deep neural networks that can be trained with them. The results presented in this paper indeed show that networks trained with such small amount of data are learning only the more common classes and have inconsistent behavior regarding the less common ones. Emergent approaches such as one-shot or few-shot learning might be able to cope with the small classes; nevertheless, the datasets remain small, unsuitable for the current trends in machine learning.

Of course it is possible to use synthetic mixtures created using isolated sound instances, such that the reference annotation is created at the same time with the audio mixture. One challenge in this is that these mixtures should be created such that they mimic real-life data, and this is not trivial. Until now, the synthetic data used in DCASE tasks was rather simplistic — for example the DCASE 2016 Task 2 synthetic audio dataset used a morphological model for creating the mixtures [50], but it was based on a very small number of event instances, while this year's Task 2 rare sound events dataset did not use any specific knowledge or rules for background and target event combinations.

Nevertheless, the topic of sound event detection attracts a lot of interest from the scientific community, and the DCASE 2017 Challenge offered an updated overview of the approaches, compared to the previous editions. For the first time, convolutional networks have dominated the methods in all three analyzed tasks, while mel representations continued to be the most commonly used features, and much of the difference in the submitted systems was down to network architecture or choice of parameters.

The role of DCASE challenge in advancing the research on sound event detection comes through successive editions bringing new and more realistic setups to the task. From detection of non-overlapping sound events in DCASE 2013 [13], to detection of overlapping sound events in synthetic mixtures and real-life recordings in DCASE 2016 [14], the challenge has evolved to presenting participants with problems that reflect real-life applications. DCASE 2017 Challenge is the first that brought the data imbalance to the task through the rare sound event detection, and the weakly-labeled data problem, and compelled participants to approach the detection task differently. The aftermath of DCASE 2017 includes research directed towards dealing with low-resource datasets, imbalanced data, and a dramatic increase in methods based on weakly-labeled data. Also, this was the first open public challenge for sound event detection on web videos, which is arguably the main source of sound events. It served as a motivation for similar tasks in following DCASE challenges and Kaggle competitions.

Feedback collected after the challenge and workshop shows that challenge participants were generally happy with the challenge organization and schedule, with most of the 42 respondents rating the organizational aspects as very good. Their motivation for participating in different tasks was diverse; some have worked on similar topics before, others considered the case of real-world data and overlapping sounds very interesting and challenging. General comments include wishes for open set classification problems, tasks geared towards wildlife preservation and bioacoustics, sound event detection in videos, spatial data, and unsupervised learning. One often encountered request is for participants to have repeated feedback about their systems in the form of a leaderboard, to use the competitive setting as a catalyst in pushing participants to further develop their systems.

VII. Future Work

The 2018 edition marks the shift to the decentralized organization, with each task being separately coordinated by one group of researchers, and very light overall coordination regarding deadlines and submission formats. DCASE 2018 Challenge had five tasks: acoustic scene classification [51], general-purpose audio tagging of Freesound content with AudioSet labels [52], bird audio detection [53], large-scale weakly labeled semi-supervised sound event detection in domestic environments [46], and monitoring of domestic activities based on multi-channel acoustics [54]. One notable evolution in DCASE 2018 Challenge is the absence of the strongly-labeled sound event detection task, according to previously presented conclusion. The current tasks reflect the interest of the individual organizer groups, and were selected through a proposal process in which the steering committee has reviewed, provided feedback and approved the tasks. With each new edition, the challenge brings new active topics and new open datasets for the research community, being an important advocate of reproducible research.

References

[1] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 1, pp. 279–288, Jan. 2016.

[2] D. Chakrabarty and M. Elhilali, "Abnormal sound event detection using temporal trajectories mixtures," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2016, pp. 216–220.

[3] D. Stowell and D. Clayton, "Acoustic event detection for multiple overlapping similar sources," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2015, pp. 1–5.

[4] S. Goetze, N. Moritz, J.-E. Appell, M. Meis, C. Bartsch, and J. Bitzer, "Acoustic user interfaces for ambient-assisted living technologies," *Inf. Health Social Care*, vol. 35, no. 3/4, pp. 125–143, 2010.

[5] E. Principi, D. Droghini, S. Squartini, P. Olivetti, and F. Piazza, "Acoustic cues from the floor: A new approach for fall classification," *Expert Syst. Appl.*, vol. 60, pp. 51–61, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417416301658

[6] G. Dekkers *et al.*, "The SINS database for detection of daily activities in a home environment using an acoustic sensor network," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Nov. 2017, pp. 32–36.

[7] N. Takahashi, M. Gygli, and L. V. Gool, "Aenet: Learning deep audio features for video analysis," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 513–524, Mar. 2018.

[8] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22st ACM Int. Conf. Multimedia*, Nov. 2014, pp. 1041–1044.

[9] A. Mesaros *et al.*, "DCASE2017 challenge setup: Tasks, datasets and baseline system," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Nov. 2017, pp. 85–92.

[10] A. S. Bregman, *Auditory Scene Analysis*. Cambridge, MA, USA: MIT Press, 1990.

[11] W. W. Gaver, "How do we hear in the world? explorations in ecological acoustics," *Ecological Psychol.*, vol. 5, no. 4, pp. 285–313, 1993.

[12] T. Heittola, E. Çakır, and T. Virtanen, *The Machine Learning Approach for Analysis of Sound Scenes and Events*. Cham, Switzerland: Springer, 2018, pp. 13–40.

[13] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Trans. Multimedia*, vol. 17, no. 10, pp. 1733–1746, Oct. 2015.

[14] A. Mesaros *et al.*, "Detection and classification of acoustic scenes and events: Outcome of the DCASE 2016 challenge," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 2, pp. 379–393, Feb. 2018.

[15] R. Stiefelhagen, K. Bernardin, R. Bowers, R. T. Rose, M. Michel, and J. Garofolo, "The clear 2007 evaluation," in *Multimodal Technologies for Perception of Humans*, R. Stiefelhagen, R. Bowers, and J. Fiscus, Eds. Berlin, Germany: Springer, 2008, pp. 3–34.

[16] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Appl. Sci.*, vol. 6, no. 6, 2016, Art. no. 162. [Online]. Available: http://www.mdpi.com/2076-3417/6/6/162

[17] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, "Context-dependent sound event detection," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, p. 1, 2013.

[18] A. Mesaros, O. Dikmen, T. Heittola, and T. Virtanen, "Sound event detection in real life recordings using coupled matrix factorization of spectral representations and class activity annotations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 151–155.

[19] J. Gemmeke, L. Vuegen, P. Karsmakers, B. Vanrumste, and H. Van hamme, "An exemplar-based NMF approach to audio event detection," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, Oct. 2013, pp. 1–4.

[20] A. Mesaros, T. Heittola, and T. Virtanen, "TUT database for acoustic scene classification and sound event detection," in *Proc. 24th Eur. Signal Process. Conf.*, 2016, pp. 1128–1132.

[21] C. Fellbaum, *WordNet: An Electronic Lexical Database*. Cambridge, MA, USA: Bradford Books, 1998.

[22] G. Forman and M. Scholz, "Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement," *SIGKDD Explorations Newslett.*, vol. 12, no. 1, pp. 49–57, Nov. 2010. [Online]. Available: http://doi.acm.org/10.1145/1882471.1882479

[23] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, New Orleans, LA, USA, 2017, pp. 776–780.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[25] H. Abdi and L. Williams, "Jackknife," *Encyclopedia of Research Design*. Newbury Park, CA, USA: Sage, 2010, pp. 1–10.

[26] H. Lim, J. Park, and Y. Han, "Rare sound event detection using 1D convolutional recurrent neural networks," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Nov. 2017, pp. 80–84.

[27] E. Cakir and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Nov. 2017, pp. 27–31.

[28] F. Vesperini *et al.*, "A hierarchic multi-scaled approach for rare sound event detection," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[29] H. Phan, M. Krawczyk-Becker , T. Gerkmann, and A. Mertins, "DNN and CNN with weighted and multi-task loss functions for audio event detection," *DCASE2017 Challenge, Tech. Rep.*, Sep. 2017.

[30] J. Wang and S. Li, "Multi-frame concatenation for detection of rare sound events based on deep neural network," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[31] J. Wang, W. Zhang, and J. Liu, "Transfer learning based DNN-HMM hybrid system for rare sound event detection," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[32] S. Adavanne and T. Virtanen, "A report on sound event detection with different binaural features," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[33] Y. Chen, Y. Zhang, and Z. Duan, "DCASE2017 sound event detection using convolutional neural network," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[34] S. Adavanne, P. Pertilä, and T. Virtanen, "Sound event detection using spatial features and convolutional recurrent neural network," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 771–775.

[35] I.-Y. Jeong, S. Lee, Y. Han, and K. Lee, "Audio event detection using multiple-input convolutional neural network," in *Proc. Detection Classif. Acoust. Scenes Events Workshop*, Nov. 2017, pp. 51–54.

[36] R. Lu and Z. Duan, "Bidirectional GRU for sound event detection," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[37] C. Kroos and M. D. Plumbley, "Neuroevolution for sound event detection in real life audio: A pilot study," *DCASE2017 Challenge, Tech. Rep.*, Nov. 2017, pp. 64–68.

[38] S. Adavanne and T. Virtanen, "Sound event detection using weakly labeled dataset with stacked convolutional and recurrent neural network," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[39] D. Lee, S. Lee, Y. Han, and K. Lee, "Ensemble of convolutional neural networks for weakly-supervised sound event detection using multiple scale input," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[40] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Surrey-CVSSP system for DCASE2017 challenge task4," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[41] J. Salamon, B. McFee, and P. Li, "DCASE 2017 submission: Multiple instance learning for sound event detection," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[42] J. Lee, J. Park, and J. Nam, "Combining multi-scale features using sample-level deep convolutional neural networks for weakly supervised sound event detection," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[43] A. Dang, T. Vu, and J.-C. Wang, "Deep learning for DCASE2017 challenge," DCASE2017 Challenge, Tech. Rep., Sep. 2017.

[44] S. Parekh, S. Essid, A. Ozerov, N. Q. Duong, P. Perez, and G. Richard, “Weakly supervised representation learning for unsynchronized audio-visual events,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops*, 2018, pp. 2518–2519.

[45] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, “Multi-level attention model for weakly supervised audio classification,” in *Proc. Detection Classification Acoust. Scenes Events 2018 Workshop*, Nov. 2018, pp. 188–192.

[46] R. Serizel, N. Turpault, H. Eghbal-Zadeh, and A. Parag Shah, “Large-scale weakly labeled semi-supervised sound event detection in domestic environments,” in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Nov. 2018, pp. 19–23.

[47] Y. Li and X. Li, “The SEIE-SCUT systems for IEEE AASP challenge on DCASE 2017: Deep learning techniques for audio representation and classification,” in *Proc. Detection Classification Acoustic Scenes Events 2018 Workshop*, Sep. 2017.

[48] A. Kumar and B. Raj, “Weakly supervised scalable audio content analysis,” in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2016, pp. 1–6.

[49] V. Morfi and D. Stowell, “Data-efficient weakly supervised learning for low-resource audio event detection using deep learning,” in *Proc. Detection Classif. Acoust. Scenes Events Workshop*, Nov. 2018, pp. 123–127.

[50] G. Lafay, M. Lagrange, M. Rossignol, E. Benetos, and A. Roebel, “A morphological model for simulating acoustic scenes and its application to sound event detection,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 24, no. 10, pp. 1854–1864, Oct. 2016.

[51] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proc. Detection Classification. Acoust. Scenes Events Workshop*, Nov. 2018, pp. 9–13.

[52] E. Fonseca *et al.*, “General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline,” in *Proc. Detection Classification Acoust. Scenes Events Workshop*, Nov. 2018, PP/99– 69–73.

[53] D. Stowell, Y. Stylianou, M. Wood, H. Pamuła, and H. Glotin, “Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge,” *Methods Ecol. Evol.*, 2018. [Online]. Available: https://arxiv.org/abs/1807.05812]

[54] G. Dekkers, L. Vuegen, T. van Waterschoot, B. Vanrumste, and P. Karsmakers, “DCASE 2018 Challenge - Task 5: Monitoring of domestic activities based on multi-channel acoustics,” KU Leuven, Leuven, Belgium, *Tech. Rep.*, 2018. [Online]. Available: https://arxiv.org/abs/1807.11246

Annamaria Mesaros received the M.Sc. and Ph.D degrees in electronics and telecommunications in 2001 and 2007, respectively, from Technical University of Cluj Napoca, Romania, and the Doctor of Science degree in signal processing from Tampere University (previously known as Tampere University of Technology, TUT), Finland, in 2012. She is currently an Assistant Professor at Tampere University (TUT). She has also been working as a Postdoctoral Researcher at Aalto University, Helsinki, Finland, within the Finnish Centre of Excellence in Computational Inference Research. Her research focuses on sound event detection in real-world multisource environments, including semantic aspects of human-generated sound annotation.

Aleksandr Diment received the M.Sc. degree in information technology from Tampere University (previously known as Tampere University of Technology, TUT), Finland, in 2013. He is currently a doctoral student at Tampere University. His doctoral thesis covers such topics as robust audio analysis, semi-supervised learning, auditory scene analysis and transfer learning. Additionally, his research interests include voice effort analysis, computational paralinguistics, prosody analysis, and music information retrieval.

Benjamin Elizalde is currently working toward the Ph.D. degree at Carnegie Mellon University, Pittsburgh, PA, USA. He received the M.Sc. degree in information technology from Instituto Tecnologico de Monterrey in 2012. He worked as a Staff Researcher at ICSI-UC Berkeley in the Audio & Multimedia Lab from 2012 to 2015. His main research interests include machine learning for audio signal processing.

Toni Heittola received the M.Sc. degree in information technology from Tampere University (previously known as Tampere University of Technology, TUT), Finland, in 2004, where he is currently working toward the Ph.D. degree. His main research interests include sound event detection in real-life environments, sound scene classification, and audio content analysis.

Emmanuel Vincent received the Ph.D. degree in music signal processing from the Institut de Recherche et Coordination Acoustique/Musique (Ircam, Paris, France) in 2004 and worked as a Research Assistant with the Centre for Digital Music at Queen Mary, University of London (United Kingdom), from 2004 to 2006. He is a Senior Research Scientist with Inria, Nancy, France. His research focuses on statistical machine learning for speech and audio signal processing, with application to audio source localization and separation, speech enhancement, and robust speech and speaker recognition. He is a Founder of the series of Signal Separation Evaluation Campaigns and CHiME Speech Separation and Recognition Challenges. He was an Associate Editor for IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

Bhiksha Raj received the Ph.D. degree in electrical and computer engineering from Carnegie Mellon, Pittsburgh, PA, USA, in 2000. He is a Professor of the Computer Science Department, Carnegie Mellon University where he leads the Machine Learning for Signal Processing Group. He joined the Carnegie Mellon faculty in 2009, after spending time at the Compaq Cambridge Research Labs and Mitsubishi Electric Research Labs. He has devoted his career to developing speech- and audio-processing technology. He has had several seminal contributions in the areas of robust speech recognition, audio content analysis and signal enhancement, and has pioneered the area of privacy-preserving speech processing. He is also the Chief Architect of the popular Sphinx-4 speech-recognition system.

Tuomas Virtanen received the M.Sc. and Doctor of Science degrees in information technology from Tampere University (previously known as Tampere University of Technology, TUT), Finland, in 2001 and 2006, respectively. He is currently a Professor at Tampere University. He is known for his pioneering work on single-channel sound source separation using non-negative matrix factorization-based techniques, and their application to noise-robust speech recognition, music content analysis, and audio event detection. In addition to the above topics, his research interests include content analysis of audio signals in general and machine learning. He has authored about 100 scientific publications on the above topics. He is a member of the Audio and Acoustic Signal Processing Technical Committee of IEEE Signal Processing Society.