Post-Processing Independent Evaluation of Sound Event Detection Systems

Janek Ebbers, Reinhold Haeb-Umbach, Romain Serizel

POST-PROCESSING INDEPENDENT EVALUATION OF SOUND EVENT DETECTION SYSTEMS

Janek Ebbers, Reinhold Haeb-Umbach

Paderborn University,
Department of Communications Engineering,
33098 Paderborn, Germany,
{ebbers,haeb}@nt.upb.de

Romain Serizel

Université de Lorraine, CNRS,
Inria, Loria,
F-54000 Nancy, France,
romain.serizel@loria.fr

ABSTRACT

Due to the high variation in the application requirements of sound event detection (SED) systems, it is not sufficient to evaluate systems only in a single operating point. Therefore, the community recently adopted the polyphonic sound detection score (PSDS) as an evaluation metric, which is the normalized area under the PSD-ROC. It summarizes the system performance over a range of operating points. Hence, it provides a more complete picture of the overall system behavior and is less biased by hyper parameter tuning. So far PSDS has only been computed over operating points resulting from varying the decision threshold that is used to translate the system output scores into a binary detection output. However, besides the decision threshold there is also the post-processing that can be changed to enter another operating mode. In this paper we propose the post-processing independent PSDS (piPSDS) which computes PSDS over operating points with varying post-processings and varying decision thresholds. It summarizes even more operating modes of an SED system and allows for system comparison without the need of implementing a post-processing and without a bias due to different post-processings. While piPSDS can in principle also combine different types of post-processing, we here, as a first step, present median filter independent PSDS (miPSDS) results for this year's DCASE Challenge Task4a systems. Source code is publicly available in our sed_scores_eval package[1].

***Index Terms*—** sound event detection, polyphonic sound detection, evaluation, post-processing, median filter

1. INTRODUCTION

Machine listening is recently attracting increased interest not only from academia but also from industry. It is the field of developing machines which can replicate the human ability of recognizing and processing a large number of different sounds. There are many sub-disciplines to machine listing, with sound event detection (SED) [1] being one of them. Its aim is to recognize, classify and temporally localize sounds within an input audio. Due to the large number of possible applications, sounds and environments, one particular challenge is that there is often no or only little training data that perfectly matches the target application. Therefore, there is a particular interest in approaches for model training which can exploit imperfect data, such as weakly labeled learning [2, 3] and/or training with mismatched or unlabeled data [4, 5], as investigated by the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge Task 4 [6] for several years now.

Another more fundamental challenge for successful SED system development is the meaningful evaluation and comparison of system performance, where the choice of the evaluation metric can have a large impact [7]. Firstly, there is the complexity of the event matching between detected and ground truth events. Currently there exist three different approaches namely segment-based, collar-based and intersection-based [8, 9]. The DCASE Challenge Task 4 recently moved to intersection-based evaluation as it is more robust w.r.t. ambiguities in the ground truth labeling. Secondly, due to the high variation in application requirements, there is often not a single optimal system behavior as, e.g., expressed by the F_1-score. In some applications, missed hits may, e.g., be much more severe than false alarms. Therefore, system evaluation must ideally represent all different operating modes equally to capture the overall system behavior. The polyphonic sound detection score (PSDS) [9, 10] has been employed to capture performance over the range of decision thresholds, which are used to translate soft system output scores[2] into binary decisions. Therefore, system comparison using PSDS is also less biased by threshold tuning w.r.t. to a certain operating point.

However, the post-processing [11] (e.g. median filtering), that is applied to the classifier output either before or after thresholding, has also a large impact on the system performance, which is mostly underinvestigated. In particular, system comparisons may be biased due to the employment of different post-processings. Similar to the decision threshold, the type and parameters of the post-processing can be understood as operating parameters of the system and may be adjusted to enter another operating mode.

In this paper we propose post-processing independent PSDS (piPSDS) which summarizes performance over both different post-processings and decision thresholds. Hence, it gives an even more complete picture of the system's performance over different operating modes and furthermore is less biased by hyper-parameter tuning. We perform investigations on this year's DCASE Challenge Task 4 submissions and show that 1) there is indeed a large impact on evaluation results due to post-processing 2) for different operating points there are different optimal post-processings and 3) the proposed piPSDS allows SED system evaluation unbiased from threshold and post-processing tuning.

The rest of the paper is structured as follows. First, we recapitulate the preliminaries of SED, its evaluation and the PSDS in Sec. 2.1, Sec. 2.2 and Sec. 2.3, respectively. Our proposed piPSDS is presented in Sec. 3. Finally, we show results in Sec. 4 and draw conclusions in Sec. 5.

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 282835863.

[1] https://github.com/fgnt/sed_scores_eval

[2] Note the ambiguity of the term score here, where PSD score refers to a metric value while output scores refer to soft class activity predictions of a model/neural network.

2. PRELIMINARIES

2.1. Sound Event Detection

To not only recognize but also temporally localize sound events, SED systems perform multi-label classification within smaller time-windows of an audio clip, e.g., at short-time Fourier transform (STFT) frame-level. For each window n a system provides soft classification scores $y_{n,c}$ for each event class c out of a set of C predefined sound event classes of interest. These scores represent the predicted activity of the event within a particular time-window. To obtain a hard decision, soft classification scores can be binarized using a certain decision threshold γ_c, where the class c is assumed active in the n-th window if $y_{n,c} \geq \gamma_c$, else it is assumed inactive. Connected active windows are then merged into a detected event $(\hat{t}_{\text{on},i}, \hat{t}_{\text{off},i}, \hat{c}_i)$ defined by onset time $\hat{t}_{\text{on},i}$, offset time $\hat{t}_{\text{off},i}$ and class label $\hat{c}_i$, respectively, where i represents the event index. Usually it is beneficial to run some kind of post-processing before or after binariazation to obtain meaningfully connected event predictions and be more robust w.r.t. outliers. Common post-processings are, e.g., median filtering [12] and Hidden Markov Model smoothing [13]. The type and hyper-parameters of the post-processing, as well as the decision threshold and any other hyper-parameters that may be easily changed during application are summarized as a system's operating parameters τ in the following.

2.2. Evaluation of Detected Events

The evaluation of the detected events of event class c for specific operating parameters τ is, in accordance with other classification tasks, based on counting the intermediate statistics $N_{\text{TP},c,\tau}$, $N_{\text{FN},c,\tau}$ and $N_{\text{FP},c,\tau}$, which refer to the numbers of

- ground truth (GT) events that have been correctly detected by the system a.k.a. true positive (TP) detections,
- GT events that have not been detected by the system a.k.a. false negative (FN) detections,
- detected events that do not match any GT event a.k.a. false positive (FP) detections,

accumulated over the whole evaluation set, respectively. Bilen et al. [9] have further taken cross triggers (CTs) into account, a.k.a. substitutions, with $N_{\text{CT},c,k,\tau}$ being the number of FPs of class c matching GT events from another event class k, which may impair user experience more than standalone FPs.

When counting above intermediate statistics, different approaches exist for the temporal matching between detected events and GT events. As the definitions of PSDS and piPSDS, however, do not depend on the temporal matching that is used, we here only briefly recap intersection-based evaluation which has recently been used for PSDS computation as it is more robust w.r.t. ambiguities in the labeling of the evaluation data. Note, however, that one could instead also compute segment-based and collar-based [8] (pi)PSDS.

Intersection-based evaluation requires detected events to intersect with GT events by at least a fraction ρ_{DTC} to be not counted as a FP detection. Moreover, it requires a GT event to intersect with non-FP events by at least a fraction ρ_{GTC} to be counted as a TP detection. Further, if an FP event intersects with a GT event of another class by at least a fraction ρ_{CTTC} it is counted as a CT.

Of particular interest are in the following the TP rate (TPR) defined as $r_{c,\tau} = \frac{N_{\text{TP},c,\tau}}{N_{\text{TP},c,\tau}+N_{\text{FN},c,\tau}}$, and the effective FP rate (eFPR)

$$e_{c,\tau} = \frac{N_{\text{FP},c,\tau}}{T_{\text{ds}}} + \alpha_{\text{CT}} \frac{1}{C-1} \sum_{\substack{k \\ k \neq c}} \frac{N_{\text{CT},c,k,\tau}}{T_k} \quad . \tag{1}$$

which consists of the FPR $\frac{N_{\text{FP},c,\tau}}{T_{\text{ds}}}$ plus an additional penalty on CT rates (CTRs) $\frac{N_{\text{CT},c,k,\tau}}{T_k}$ averaged over all other classes $k \neq c$ and weighted by α_{CT}. Note that, with intersection-based evaluation, there is not a countable number of negative events, which is why the FPR is computed w.r.t. the total duration of the evaluation dataset T_{ds}, whereas CTRs are computed w.r.t. the total duration of activity T_k of the k-th class within the evaluation dataset.

2.3. Polyphonic Sound Detection Score

To compute PSDS [9], one starts with the computation of single-class PSD-ROC curves $r_c(e)$ for each event class c. $r_c(e)$ is obtained as a continuous "staircase-type" interpolation of true positive rates $r_{c,\tau}$ plotted over corresponding eFPRs $e_{c,\tau}$ for different operating parameters $\tau \in \widehat{\mathcal{T}}_c$.

While τ may be any (set of) hyper-parameter(s) that may change system behavior, it has so far, in accordance with the standard definition of ROC curves [14], only been considered to be the decision threshold used to translate soft prediction scores into binary detections. Here, an algorithm for the efficient joint evaluation of all possible decision thresholds has been proposed in [10]. Note that, in contrast to standard ROC curves, it is here not always guaranted that $r_{c,\tau}$ is monotonically increasing with $e_{c,\tau}$, when, e.g., sophisticated intersection-based evaluation is employed. As in operation, however, one would always prefer the operating point with a higher true positive rate at lower or equal false positive rate if available, $\widehat{\mathcal{T}}_c$ represents only best case operating parameters:

$$\widehat{\mathcal{T}}_c = \left\{\tau \middle| \nexists\, \lambda \text{ with } e_{c,\lambda} \leq e_{c,\tau} \text{ and } r_{c,\lambda} > r_{c,\tau}\right\}. \tag{2}$$

Having the single-class PSD-ROC curves $r_c(e)$, the overall PSD-ROC curve is defined as the effective true positive rate

$$r(e) = \mu_{\text{TP}}(e) - \alpha_{\text{ST}}\sigma_{\text{TP}}(e) \tag{3}$$

which is average per-class true positive rate minus a penalty on standard deviation over classes weighted by a metric parameter α_{ST} with

$$\mu_{\text{TP}}(e) = \frac{1}{C}\sum_{c=1}^{C} r_c(e); \quad \sigma_{\text{TP}}(e) = \sqrt{\frac{1}{C}\sum_{c=1}^{C}(r_c(e) - \mu_{\text{TP}}(e))^2}.$$

Finally, the PSDS is the normalized area under the PSD-ROC:

$$\text{PSDS} = \frac{1}{e_{\max}} \int_0^{e_{\max}} r(e)de \tag{4}$$

with the maximal false positive rate $e_{\max}$ being a metric parameter, which controls up to which false positive rate the operating points may still be relevant.

3. POST-PROCESSING INDEPENDENT POLYPHONIC SOUND DETECTION SCORE

Besides the decision threshold there is also the post-processing that we could change to enter another operating mode. As an example, Fig. 1 shows the single-class PSD-ROC curves for "Speech" from this year's "Baseline_BEATS" system [15] when using post-processing median filtering with lengths of $0.1\,\text{s}$ and $1.0\,\text{s}$, respectively. It appears that when the system is operated in low eFPR mode, than it is better to use the larger median filter window size. When the system should be operated in high TPR mode, it is better to use a smaller window size. Thus, it is reasonable and also fairly easy to choose the post-processing depending on the requirements of a given application. To account for this in the system evaluation, which is supposed to capture overall system behavior, we propose

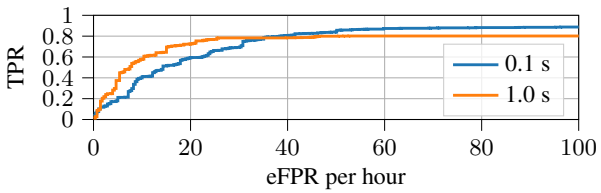

Figure 1: Baseline Speech ROCs with different median filters

to incorporate the variation of post-processing into the computation of the PSDS to get a post-processing independent PSDS (piPSDS).

To do so, we consider the operating parameters $\tau = (l, \gamma)$ to be a tuple of the post-processing l and the decision threshold γ. Here, l defines which postprocessing is used out of a predefined set of L possible post-processings. Post-processing is assumed to be applied before the binarization with the decision threshold γ, as this still allows to use the algorithm from [10] to efficiently compute the true positive and eFPRs for the whole continuous range of γ. Here, it is worth noting that, e.g., median filtering and thresholding are permutation invariant, i.e., applying the median filter before binarization yields the same result as applying it afterwards.

The definition of the PSD-ROC according to Sec. 2.3 with $\tau \in \mathbb{L} \times \mathbb{R}$, where $\mathbb{L} = \{l \in \mathbb{N} \,|\, l < L\}$, gives us the pi-PSD-ROC. Due to the restriction to best case operating points in Eq. 2 the single-class pi-PSD-ROCs can be computed as

$$r_c(e) = \max_l r_{c,l}(e) \tag{5}$$

where $r_{c,l}(e)$ represents the single-class PSD-ROC for a single post-processing l resulting only from variation of the decision threshold. Hence, the single-class pi-PSD-ROC chooses, for a given eFPR e, the post-processing with the highest TPR. piPSDS is then, analogously to Eq. 4, the normalized area under the pi-PSD-ROC.

Overall, piPSDS has two major advantages over only threshold-independent PSDS. Firstly, it better captures real-world SED applications, where it is natural to choose the post-processing that best suits the current application requirements. Secondly, for research it allows for system comparison without a bias being introduced by different post-processings.

4. RESULTS

Investigations are done with the baseline and submissions of this year's DCASE challenge Task4a. Participants have been asked to, in addition to their post-processed submission, also share the raw prediction scores as provided by their model/neural network without any further post-processing. This allows us to investigate 1) the impact of the post-processing, 2) post-processing independent evaluation. All following evaluations are performed on the DESED [16] public eval set, which is a part of the challenge evaluation data.

There are two intersection-based PSDS evaluated in the challenge, which refer to different scenarios. PSDS1 ($\rho_{\text{DTC}} = 0.7$, $\rho_{\text{GTC}} = 0.7$, $\alpha_{\text{CT}} = 0$, $\alpha_{\text{ST}} = 1$, $e_{\max} = 100/\text{hour}$) particularly evaluates the model's capability of temporally localizing sound events, whereas PSDS2 ($\rho_{\text{DTC}} = 0.1$, $\rho_{\text{GTC}} = 0.1$, $\rho_{\text{CTTC}} = 0.3$, $\alpha_{\text{CT}} = 0.5$, $\alpha_{\text{ST}} = 1$, $e_{\max} = 100/\text{hour}$) is more focused on evaluating the reliable recognition of event classes within an audio clip. Due to space constraints and with post-processing being particularly relevant for the temporal localization of sound events, we only consider PSDS1 evaluation in the following.

With median filtering being the most popular type of post-processing for SED systems, we here consider median filter independent PSDS (miPSDS) as an instance of piPSDS, where the set

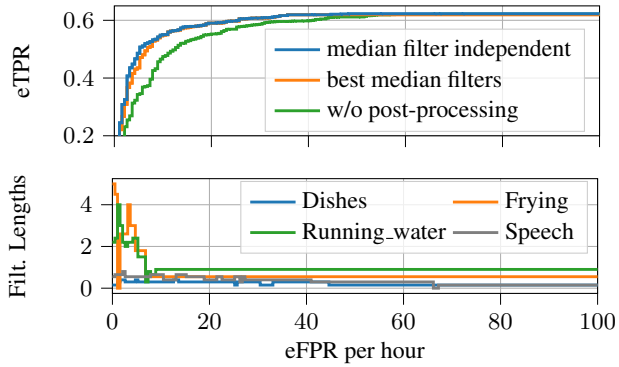

Figure 2: Upper Plot: PSD-ROCS for different post-processing setups. Lower Plot: Optimal median filter lengths over operating points as tracked by median filter independent PSD-ROC.

of possible post-processings consists of median filters with different filter lengths. As the set of median filter lengths we use 21 filter lengths linearly spaced from 0.0 s (no filtering) to 1.0 s, 10 from 1.1 s to 2.0 s, 5 from 2.2 s to 3.0 s and 4 from 3.5 s to 5.0 s overall totaling 40 different filter lengths. The implementation of the median filter equals a time continuous filtering of a piece-wise constant signal that is defined by the timestamped prediction scores submitted by the participants. This way it is ensured the systems employ the very same post-processing regardless of the system's output resolution which may vary across systems. Implementations of the median filter, miPSDS and piPSDS, with the latter taking any list of differently post-processed scores, are publicly available in the sed_scores_eval package[1], that is, in accordance with the challenge, used for evaluation.

We first run investigations on the baseline system Baseline_BEATS [15] (Baseline). In the upper subplot of Fig. 2 we compare the following PSD-ROCs:

1. median filter independent: as defined in Eq. 3
2. best median filters: choosing best performing median filter per class as follows
   $\tilde{r}_c(e) = r_{c,b}(e)$ with $b = \text{argmax}_l \text{auc}(r_{c,l}(e))$,
3. without any post-processing.

It can be seen, that by applying (best) median filtering the PSD-ROC can be significantly improved over the unprocessed case. It can be further observed, that there are operating points, especially for low eFPRs, where the mi-PSD-ROC (mi-PSD-ROC) is higher than best median filter PSD-ROC. This indicates that best median filters are, although giving best overall performance, not the best choice for each individual operating point and better performance can be achieved by choosing operating point dependent filter lengths as the mi-PSD-ROC does. In the lower subplot of Fig. 2 we plot, for some event classes, the optimal filter lengths over operating points. We can see that for lower eFPRs optimal median filters tend to be longer than for higher eFPRs, which can be explained by the fact that longer median filters better suppress short duration FPs. Further, event classes with longer per-event durations, such as "Frying" and "Running Water", tend to have overall longer median filters than short duration event classes, which makes intuitively sense.

Next, we evaluate challenge submissions[3] with, without and independent of post-processing. As submitting unprocessed scores was optional, we evaluate only systems from the 12 teams that

[3]Submissions will be made publicly available on zenodo soon with link being added in the camera-ready version in case of acceptance

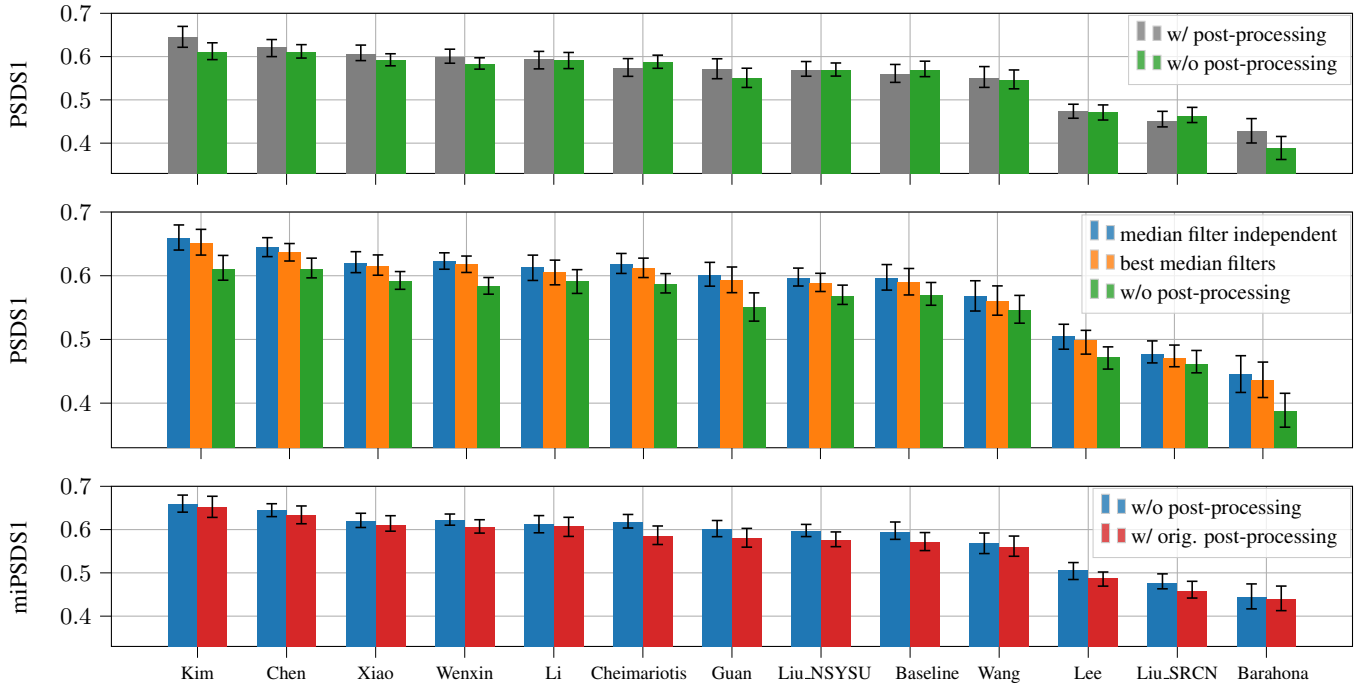

Figure 3: System Evaluation. Upper Plot: original post-processing vs. no post-processing. Middle Plot: miPSDS vs. PSDS with optimal median filter lengths per class vs. no post-processing. Lower Plot: miPSDS computed with unprocessed vs. post-processed data.

did provide them. We limit evaluation to the one single-model system per team that gave best PSDS1 performance in the challenge (with original post-processing). These systems are Barahona-AUDIAS-2 [17], Cheimariotis-DUTH-1 [18], Chen-CHT-2 [19], Guan-HIT-3 [20], Kim-GIST-HanwhaVision-2 [21], Lee-CAUET-1 [22], Li-USTC-6 [23], Liu-NSYSU-7 [24], Liu-SRCN-4 [25], Wang-XiaoRice-1 [26], Wenxin-TJU-6 [27], Xiao-FMSG-4 [28].

To be able to evaluate the variance of system performance over different runs of system training, participants submitted prediction scores for three runs of training for each system. To further track variance of results due to variations in the evaluation data, we perform bootstrapped evaluation, where evaluation is performed on 20 different $80\,\%$ fractions of the eval data. In total we evaluate $3 \cdot 20 = 60$ different setups and report the mean and $5\,\% - 95\,\%$ confidence interval of the system's performances. This evaluation procedure is the same as we used for official challenge evaluation.

We first want to investigate the impact of the post-processing on the systems' performances in the upper subplot of Fig. 3. by comparing the performance with and without the post-processing as used by the participants. It appears that for some systems, e.g., Kim-GIST-HanwhaVision-2, the performance significantly degrades when removing the post-processing, whereas for other systems the performance does not degrade or even improves. When evaluating the unprocessed scores, the ranking also changes at multiple positions to Kim, Chen, Li, Xiao, Cheimariotis, Wenxin, Baseline, Liu_NSYSU, Guan, Wang, Lee, Liu_SRCN, Barahona. This suggests that there is some bias introduced by the post-processing, particularly, whether a sophisticated post-processing is employed or not. To some extent, however, it may also be a system property that it can benefit from post-processing more than other systems.

We next evaluate our proposed miPSDS and compare it to "no processing" and "best median filters" in the middle subplot of Fig. 3. It can be seen that for all systems performance can be improved by best median filters and further improved by operating point specific median filters as considered by miPSDS. Some systems, e.g., Kim and Barahona, benefit more from best median filters / median filter independent evaluation than others, which can be explained by our previous assumption that the effectiveness of post-processing is to some extent also a system property. Here, miPSDS evaluation gives again a different ranking which is Kim, Chen, Wenxin, Xiao, Cheimariotis, Li, Guan, Liu_NSYSU, Baseline, Wang, Lee, Liu_SRCN, Barahon.

Note, that it is still possible to run additional post-processing before piPSDS evaluation to improve performance. However, it can be assumed that the possible gain is rather small and it is more likely that an additional post-processing degrades piPSDS. To investigate this, we compare miPSDS evaluated on unprocessed scores vs. scores with participants' original post-processing in the lower subplot of Fig. 3. It can be seen that in all cases the additional post-processing degrades miPSDS performance.

5. CONCLUSIONS

Due to the high variation of SED system application requirements, SED evaluation has to capture the overall system behavior over various operating points. Therefore, the community recently moved to decision threshold independent evaluation using PSDSs to capture performance over different decision thresholds used for binarization of system output scores. In this paper we proposed piPSDS which further evaluates performance over different post-processings and effectively choosing the post-processing that is best suited for a certain operating mode. It has been shown that piPSDS indeed overcomes the bias introduced due to different post-processings but still accounts for system-specific effectiveness of post-processing. It further allows for system comparison without the need of employing a sophisticated post-processing, e.g., during system development.

6. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[2] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2019, pp. 31–35.

[3] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Weakly-supervised sound event detection with self-attention," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 66–70.

[4] L. JiaKai, "Mean teacher convolution system for dcase 2018 task 4," Detection and Classification of Acoustic Scenes and Events Challenge, Tech. Rep., September 2018.

[5] J. Ebbers and R. Haeb-Umbach, "Pre-training and self-training for sound event detection in domestic environments," DCASE2022 Challenge, Tech. Rep., June 2022.

[6] DCASE 2023 Challenge Task 4a description. [Online]. Available: https://dcase.community/challenge2023/task-sound-event-detection-with-weak-labels-and-synthetic-soundscapes

[7] G. Ferroni, N. Turpault, J. Azcarreta, F. Tuveri, R. Serizel, Ç. Bilen, and S. Krstulović, "Improving sound event detection metrics: insights from dcase 2020," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2021, pp. 631–635.

[8] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences*, vol. 6, no. 6, p. 162, 2016.

[9] Ç. Bilen, G. Ferroni, F. Tuveri, J. Azcarreta, and S. Krstulović, "A framework for the robust evaluation of sound event detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 61–65.

[10] J. Ebbers, R. Haeb-Umbach, and R. Serizel, "Threshold independent evaluation of sound event detection scores," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 1021–1025.

[11] E. Benetos, D. Stowell, and M. D. Plumbley, "Approaches to complex sound scene analysis," *Computational analysis of sound scenes and events*, pp. 215–242, 2018.

[12] E. Cakir, T. Heittola, H. Huttunen, and T. Virtanen, "Polyphonic sound event detection using multi label deep neural networks," in *Proc. International Joint Conference on Neural Networks*, 2015, pp. 1–7.

[13] S. Cornell, M. Olvera, M. Pariente, G. Pepe, E. Principi, L. Gabrielli, and S. Squartini, "Domain-adversarial training and trainable parallel front-end for the dcase 2020 task 4 sound event detection challenge," in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2020.

[14] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proc. 23rd international conference on Machine learning*. ACM Press, 2006, pp. 233–240.

[15] DCASE 2023 Challenge Task 4a baseline. [Online]. Available: https://github.com/DCASE-REPO/DESED_task/tree/master/recipes/dcase2023_task4_baseline

[16] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Proc. Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[17] S. Barahona, D. de Benito-Gorron, S. Segovia, D. Ramos, and D. Toledano, "Optimizing multi-resolution conformer and crnn models for different PSDS scenarios in DCASE challenge 2023 task 4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[18] G.-A. Cheimariotis and N. Mitianoudis, "Sound event detection of domestic activities using frequency dynamic convolution and BEATS embeddings," DCASE2023 Challenge, Tech. Rep., June 2023.

[19] W.-Y. Chen, C.-L. Lu, H.-F. Chuang, Y.-H. C. Cheng, and B.-C. Chan, "Sound event detection system using pre-trained model for dcase 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[20] Y. Guan and Q. Shang, "Semi-supervised sound event detection system for DCASE 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[21] J. W. Kim, S. W. Son, Y. Song, . Kim, Hong Kook1, I. H. Song, and J. E. Lim, "Semi-supervised learning-based sound event detection using frequency dynamic convolution with large kernel attention for DCASE challenge 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[22] S. Lee, N. Kim, J. Lee, C. Hwang, S. Jang, and I.-Y. Kwak, "Sound event detection using convolution attention module for DCASE 2023 challenge task4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[23] K. Li, P. Cai, and Y. Song, "Li USTC team's submission for DCASE 2023 challenge task4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[24] C.-C. Liu, T.-H. Kuo, C.-P. Chen, C.-L. Lu, B.-C. Chan, Y.-H. Cheng, and H.-F. Chuang, "Cht+nsysu sound event detection system with pretrained embeddings extracted from beats model for dcase 2023 task 4," DCASE2023 Challenge, Tech. Rep., June 2023.

[25] M. Chen, Y. Jin, J. Shao, Y. Liu, B. Peng, and J. Chen, "DCASE 2023 challenge task4 technical report," DCASE2023 Challenge, Tech. Rep., June 2023.

[26] Y. Wang, H. Dinkel, Z. Yan, J. Zhang, and Y. Wang, "Pepe: Plain efficient pretrained embeddings for sound event detection," DCASE2023 Challenge, Tech. Rep., June 2023.

[27] X. Duo, Wenxin1 Fang and J. Li, "Semi-supervised sound event detection system for DCASE 2023 task4a," DCASE2023 Challenge, Tech. Rep., June 2023.

[28] Y. Xiao, T. Khandelwal, and R. K. Das, "FMSG submission for DCASE 2023 challenge task 4 on sound event detection with weak labels and synthetic soundscapes," DCASE2023 Challenge, Tech. Rep., June 2023.