ICA 2022
24th INTERNATIONAL CONGRESS ON ACOUSTICS
October 24(Mon) - 28(Fri), 2022 Gyeongju, Korea

PROCEEDINGS of the
24th International Congress on Acoustics

October 24 to 28, 2022 in Gyeongju, Korea

On choosing decision thresholds for anomalous sound detection in machine condition monitoring

Kevin WILKINGHOFF[1], Alessia CORNAGGIA-URRIGSHARDT[2]

[1]Fraunhofer FKIE, Germany, kevin.wilkinghoff@fkie.fraunhofer.de

[2]Fraunhofer FKIE, Germany, alessia.cornaggia-urrigshardt@fkie.fraunhofer.de

ABSTRACT

Most anomalous sound detection (ASD) systems output a score for each audio sample presented to the system. Ideally, these anomaly scores differ for normal and anomalous samples such that one can determine whether a given sample is normal or anomalous by comparing the scores to predefined thresholds. However, determining these thresholds is non-trivial, especially when no anomalous samples are provided as training data. In this work, several methods for finding such decision thresholds are evaluated and compared to each other when acoustically monitoring the condition of machines in noisy environments. To this end, the state-of-the-art in ASD for machine condition monitoring will be reviewed first. Using a state-of-the-art ASD system, experimental evaluations are conducted on the DCASE 2020 ASD dataset to evaluate differently attained decision thresholds.

Keywords: anomalous sound detection, decision threshold, machine listening

1 INTRODUCTION

Anomaly detection [1, 2] is the task of identifying samples substantially differing from normal samples that are frequently encountered. Collecting these *anomalous* samples is difficult because by definition anomalies occur only rarely and often are very costly to produce artificially. For example, when acoustically monitoring the condition of machines, creating anomalous samples of machine sounds translates to damaging potentially costly machines in very specific ways whereas recording fully functioning machines is much less costly. Furthermore, for many applications anomalies are very diverse making it practically impossible to sufficiently cover the space of anomalous samples by collecting as many as possible of them. Hence, in many cases anomaly detection takes place in a semi-supervised setting meaning that only normal samples are available for training a system.

Evaluating and comparing different systems for anomaly detection or sound detection should be independent of the choice of decision thresholds to allow for a more objective comparison [1, 3]. Because of this, metrics such as the area under the receiver-operating characteristic curve (ROC-AUC) that do not utilize any decision threshold are usually used. However, when setting up a system for practical applications detection thresholds are still needed to distinguish between normal and anomalous test samples. But without access to anomalous training samples, it is impossible to determine decision thresholds by simply testing multiple values and picking the best-performing one. Hence, estimating these thresholds is highly non-trivial and requires sophisticated techniques.

Anomalous sound detection (ASD) for machine condition monitoring is highly promoted through the annual DCASE challenge [4, 5, 6]. The baseline systems of the ASD tasks utilize the 90th percentile of a gamma distribution estimated from the histogram of the anomaly scores belonging to the normal training samples as a decision threshold. For all systems that participated in the DCASE challenge 2021 either no procedure for automatically estimating the decision thresholds is explicitly mentioned or the same (or a very similar) procedure as used by the baseline system is applied [7, 8, 9, 10, 11, 12]. The most likely reason for a lack of focus on techniques for choosing decision thresholds is that the evaluation of the ASD systems is based on the AUC

score and thus no decision thresholds are needed. This is done to have an objective comparison between the ASD performance of the systems and to prevent participants from cheating by utilizing anomalous samples of the development set for estimating thresholds. Furthermore, challenges are usually carried out only for research purposes without the goal of obtaining a fully functioning ASD system for a real-world application that would inevitably need a sophisticated technique for determining a decision threshold.

The goal of this work is to investigate multiple techniques for estimating decision thresholds in the context of anomalous sound detection for machine condition monitoring. For this purpose, first the state-of-the-art in ASD including a specific system for experimental evaluations is briefly reviewed. Second, multiple methods for estimating a decision threshold are presented. In experimental evaluations on the DCASE 2020 ASD dataset [4], these techniques are applied and compared to each other.

2 STATE-OF-THE-ART OF ANOMALOUS SOUND DETECTION

2.1 Review

First, the state-of-the-art in ASD will be reviewed. For this purpose, we will mainly follow [5]. There are two general state-of-the-art ASD paradigms for machine condition monitoring. Both rely on deep learning. The first one consists of using an autoencoder trained on normal data only. Here, it is assumed that the autoencoder can reconstruct normal data better than anomalous data due to deviations from the normal data used for training the model and thus the reconstruction error can be used as an anomaly score. Many different autoencoder architectures have been used for this purpose, e.g. class-conditioned autoencoders [13] or group masked autoencoders [14]. This approach of directly estimating the distribution of normal data is also more generally called inlier modelling (IM).

The second approach is to train a discriminative model to learn meaningful embeddings of the data. Here, it is assumed that the information needed to discriminate among predefined classes also captures the information needed to detect anomalous samples. This approach is called outlier exposure (OE) [15]. In machine condition monitoring, most models are trained to discriminate among different machine types or even finer subdivisions of the data such as different machine states or noise types [11]. To train an OE model, usually angular margin losses such as ArcFace [16] or AdaCos [17] are used [18, 19]. These losses ensure that not only inter-class similarity is minimized but simultaneously maximize intra-class similarity using an angular margin in combination with the cosine distance. Thus after training, embeddings belonging to normal samples of a specific class are concentrated around a learned mean embedding and anomalous samples are expected to have a larger angle than normal samples to this mean enabling the detection of anomalies.

Many state-of-the-art systems utilize both ASD paradigms. As noted in [5], there are two different ways to combine both approaches: a parallel and a sequential approach. The parallel approach is simply an ensemble of multiple OE and IM models [20, 21, 22] and the sequential approach consists of first applying an OE model as a feature extractor and then using an IM model for these features [23, 11]. Compared to a parallel approach, a sequential approach has the advantage that the system consists of fewer hyperparameters. However, when training a discriminative model to extract features some information needed to detect anomalous data may be lost if this information is not important for identifying the pre-defined classes.

2.2 Used system

For all experimental evaluations in this work, the system presented in [24] is used. The system is a sequential approach consisting of a neural network trained to extract discriminative audio embeddings from log-mel spectrograms using the sub-cluster AdaCos loss and a GMM for IM. The sub-cluster AdaCos loss is an extension of the AdaCos loss specifically designed for ASD. This means that the loss is also an angular margin loss with an adaptive scale parameter. The major difference to the standard AdaCos loss is that instead of learning a single class center for each class, the loss learns multiple sub-clusters for each class to learn more complex distributions. In this case, the classes are defined as specific machines recorded in noisy environments (see Section 4.1). More details about the sub-cluster AdaCos loss can be found in [24]. For all experiments, 32 sub-clusters for each class are used. When computing the log-mel spectrograms 128 mel bins, a window size of 1024 and a hop size of 512 are used.

Table 1. Modified ResNet architecture used for extracting discriminative embeddings.

layer name	structure	output size
input	-	313×128
2D convolution	7×7, stride$= 2$	$157 \times 64 \times 16$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$78 \times 31 \times 16$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$39 \times 16 \times 32$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$20 \times 8 \times 64$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$10 \times 4 \times 128$
max pooling	10×1, stride$= 1$	4×128
flatten	-	512
dense (representation)	linear	128
sub-cluster AdaCos	32 sub-clusters	41

The neural network has a modified ResNet architecture [25] as shown in Table 1 and is implemented in Tensorflow [26]. The model is trained for 400 epochs with a batch size of 64 using Adam [27]. For data augmentation, mixup [28] with a uniformly sampled mixing coefficient is used to randomly generate additional data and prevent overfitting of the model. After training, the model is used to extract embeddings, which are length-normalized by projecting them to the unit sphere. The only exception is the machine type "ToyConveyor" for which the temporal means of the log-mel spectrograms are used instead of the embeddings because these representations yield a much better ASD performance for this machine type [24]. For each pre-defined class i.e. for each specific machine of a given machine type, the distributions of the resulting embeddings are then estimated with Gaussian mixture models (GMMs) with 32 Gaussian components and a full covariance matrix, which is regularized by adding 0.001 to the diagonal as implemented in scikit-learn [29]. To obtain anomaly scores, the corresponding log-likelihood values of the GMMs are used.

3 FINDING DECISION THRESHOLDS

There are many different approaches for finding decision thresholds [30, 31]. For arbitrary anomaly scores obtained with supervised classifiers potentially being biased, these anomaly scores can be calibrated by converting them into (pseudo-)probabilities for which thresholds can be determined [32, 33]. Usually, a probability of 0.5 is used as a threshold for these calibrated scores. Since the anomaly scores of the used system already are log-likelihood values, a mapping to probabilities is not necessary. For multivariate data or scores, a threshold can be determined by using the empirical distribution function of the squared Mahalanobis distance and using a critical value such as a small quantile of the chi-squared distribution, which is the theoretical distribution function, as a threshold. [34]. An extension of this approach uses an adaptive threshold [35]. However, anomaly scores are usually univariate and thus multivariate approaches are not suitable. When continuously monitoring data streams for anomalies, these data streams themselves consist of sequential samples of which most samples are normal and only a few are anomalous. Therefore, thresholds can utilize previously encountered samples and need to adapt to changes occurring in the data stream [36, 37, 38]. This is very different from the ASD setting investigated in this work where individual recordings are either entirely normal or anomalous.

Now, several techniques for automatically estimating decision thresholds using only scores obtained with normal data will be reviewed. The general idea of all methods is to estimate a threshold that separates the extreme values of the training scores from the rest. Most of these methods are based on the assumption that the considered data, i.e. the anomaly scores, follow some specific distribution, typically a normal distribution. As this is not true for anomaly scores in general, the considered methods may work only to some extent.

3.1 Gamma distribution percentile (GP)

The strategy used by the baseline systems of the DCASE challenge [5, 6] is to fit a gamma distribution to the scores obtained with the normal training samples and use the inverse of the 90th percentile of the cumulative distribution function as the decision threshold. Test scores larger than this threshold are marked as anomalous; otherwise they are considered normal.

3.2 Histogram percentile (HP)

One can also directly use the histogram of the scores without fitting a distribution first. Note that this silently assumes a uniform distribution. We used the 90th percentile of the histogram of the scores as the decision threshold as done in [11].

3.3 Standard Deviation (SD)

One of the most commonly used approaches is to fit a normal distribution to the scores. All values exceeding the range $\mu \pm \alpha * \sigma$ are marked as anomalous, where μ and σ are the mean and standard deviation of the normal scores. Note, that technically this implies the usage of two thresholds. Since the anomaly scores in this work consist of negative log-likelihoods and thus scores belonging to normal and anomalous samples are assumed to be linearly separable, only the upper threshold is used. To have a consistent evaluation with the previous two approaches, we used $\alpha = 1.28$, which approximately corresponds to the 90th percentile.

3.4 Median Absolute Deviation (MAD)

Following the assumption that the median is more robust against outliers than the mean, decision thresholds may be obtained by $\widetilde{\mathrm{x}} \pm \alpha * \mathrm{MAD}$, where MAD is given by $\mathrm{MAD} = \beta * \mathrm{median}(|\mathrm{x} - \widetilde{\mathbf{x}}|)$ and $\widetilde{\mathbf{x}}$ is the median value of the score values x. [31] proposes $\alpha = 3$ and $\beta = 1.4826$ following [39], [30] uses $\alpha = 2$, which is also the value we used.

3.5 Interquartile Range (IQR)

This approach is based on the division of the score values x into subsets by setting Q1 and Q3 such that $\mathrm{x} \geq \mathrm{Q1}$ for 75% and $\mathrm{x} \geq \mathrm{Q3}$ for 25% of x. Then $\mathrm{IQR} = \mathrm{Q3} - \mathrm{Q1}$. Values outside the range $\mathrm{Q1} - \alpha * \mathrm{IQR}$ and $\mathrm{Q3} + \alpha * \mathrm{IQR}$ are considered anomalous. Typically, $\alpha = 1.5$ is assumed [39]. We used, $\alpha = 0.5$ as this significantly improved the performance. This approach is also known as *boxplot* [30].

3.6 One-class support vector machine (OCSVM)

To estimate the support of a distribution, a one-class support vector machine [40], which learns to discriminate between regions of high and low density using a hypersphere in high-dimensional space, can also be used. We used the implementation of scikit-learn [29] with a linear kernel. For the hyperparamter ν, we used a value of 0.1 i.e. 10% of the normal training scores are treated as anomalous.

3.7 Generalized Extreme Studentized Deviate (GESD)

GESD [41] is an iterative approach based on the Grubbs's test (GRUBBS) [42]. This statistical test, named after Grubbs , assumes a normal distribution and is calculated on the so-called Grubbs statistic

$$G = \frac{|\max(\mathrm{x}) - \mu|}{\sigma} \tag{1}$$

with mean μ and standard deviation σ. G is evaluated against a critical value of the student's t-distribution with a significance level α, set to 0.05 as default, and data size N:

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}}. \tag{2}$$

GRUBBS only tests for a single anomalous sample. For GESD, GRUBBS is thus repeated iteratively until no further anomalies are detected.

3.8 Clever Standard Deviation (cleverSD)

CleverSD [43] is another iterative approach. The idea is to repeatedly eliminate a single sample with the highest score from the training scores in case it is found to be anomalous by applying SD ($\alpha = 2$). This is done until no additional anomaly is found. We used the last score removed by this approach as the decision threshold.

3.9 Two-stage Thresholding (-x2)

Generalizing cleverSD, [31] suggests yet another iterative anomaly detection method called multi-stage thresholding. The main idea is to simply apply a non-iterative method multiple times to remove anomalies from the training scores. The difference to cleverSD is, that not only one anomaly but all anomalies detected are removed in each iteration. Experiments have shown that two stages are sufficient and thus we only used two iterations. When applying this technique to non-iterative approaches, we use the name of the method with the suffix -x2 to denote its two-stage version.

4 EXPERIMENTS

4.1 Dataset

For all experiments in this work, the DCASE 2020 ASD dataset [4] has been used. It consists of recordings from six different machine types, namely "ToyCar" and "ToyConveyor" from ToyAdmos [44], and "fan", "pump", "slider" and "valve" from MIMII [45]. Each recording contains a specific machine sound as well as factory noise and has a length of 10 seconds with a sampling rate of 16 kHz. There are six to seven different machine ids per machine type that correspond to a specific machine of that type and a total of 42 machine ids. These machine ids are used as classes when training the discriminative model described in Section 2.2.

The dataset is divided into a training set, a development set and an evaluation set. The training set consists of approximately 1000 normal sounds for each machine id. The development set consists of 100 to 200 normal sounds and 100 to 200 anomalous sounds for each of one half of the machine ids belonging to each machine type. The evaluation set consists of approximately 400 recordings containing both normal and anomalous sounds for each of the other half of machine ids.

4.2 Comparison of the decision methods

The scores obtained with the normal training data are used to estimate thresholds for ASD scores for each machine type and each machine id individually. These thresholds or the models representing them are evaluated both on the development and the evaluation set by applying them to the corresponding test sets containing a mixture of normal and anomalous samples. F1 scores are then calculated for each machine id individually and the mean is calculated for each machine type. For comparison, the performance obtained with a single optimal threshold is evaluated as an additional method denoted by *optimum*. These optimal thresholds are calculated by simply trying multiple values as decision thresholds, calculating the corresponding F1-scores and denoting the highest achieved F1-score for each machine type. To have a more robust estimation of all results, the ASD system is trained five times and the whole evaluation procedure is repeated five times for each method. Then, the mean of the five resulting F1 scores is calculated as the final performance. The final results for development and evaluation set are listed in Table 2 and Table 3, respectively. The best method for estimating decision thresholds per machine type is underlined. Additionally, average F1 scores computed over all machines types are provided. The results show that different threshold detection methods yield varying performances for distinct machine types.

To compare the different methods while reducing the influence of the difference in performance for individual machine types, we used the ratio between F1-score of a method and the best possible F1-score obtained with a single threshold, i.e. the results obtained with *optimum*, instead of the F1-scores themselves. Therefore, these values show how close an estimated threshold is to the optimal threshold and allows a better comparison between the methods regardless of the actual ASD performance of the used system for different machine types.

The results are depicted in Figure 1. The following observations can be made. First, most approaches result in a very similar ASD performance. The only exceptions are GESD, which performs worse on both the development and evaluation set, and GP/GPx2, which performs slightly worse on the development set, than

Table 2. Mean of F1 scores among all machine ids belonging to single machine types obtained with five independent trials on the development dataset. Highest F1 score among different methods for each machine type is underlined.

	fan	pump	slider	ToyCar	ToyConveyor	valve	mean
GP	0.83204	0.82253	0.91109	0.80763	0.61818	0.87869	0.81169
HP	0.75288	0.83735	0.95056	0.84823	0.67279	0.89828	0.82668
SD	0.75055	0.83927	0.95147	0.85007	0.67236	0.89985	0.82726
MAD	0.76191	0.84408	0.95138	0.85412	0.66953	0.90171	0.83046
IQR	0.78953	0.84310	0.93899	0.83723	0.67722	0.89694	0.83050
OCSVM	0.75288	0.83735	0.95049	0.84907	0.67265	0.89794	0.82673
GESD	0.66239	0.81641	0.96752	0.86416	0.59373	0.86991	0.79569
cleverSD	0.83559	0.83511	0.91850	0.82063	0.66289	0.88529	0.82633
GPx2	0.86353	0.80954	0.88840	0.77259	0.61148	0.86485	0.80173
HPx2	0.81633	0.83612	0.92617	0.81426	0.66316	0.88786	0.82398
SDx2	0.82656	0.83333	0.91861	0.80603	0.65868	0.88201	0.82087
MADx2	0.79681	0.84887	0.94039	0.84690	0.67556	0.89848	0.83450
IQRx2	0.83306	0.83445	0.91598	0.80581	0.65856	0.88336	0.82187
OCSVMx2	0.81573	0.83616	0.92650	0.81474	0.66307	0.88739	0.82393
optimum	0.92574	0.88895	0.98461	0.89175	0.68857	0.91896	0.88310

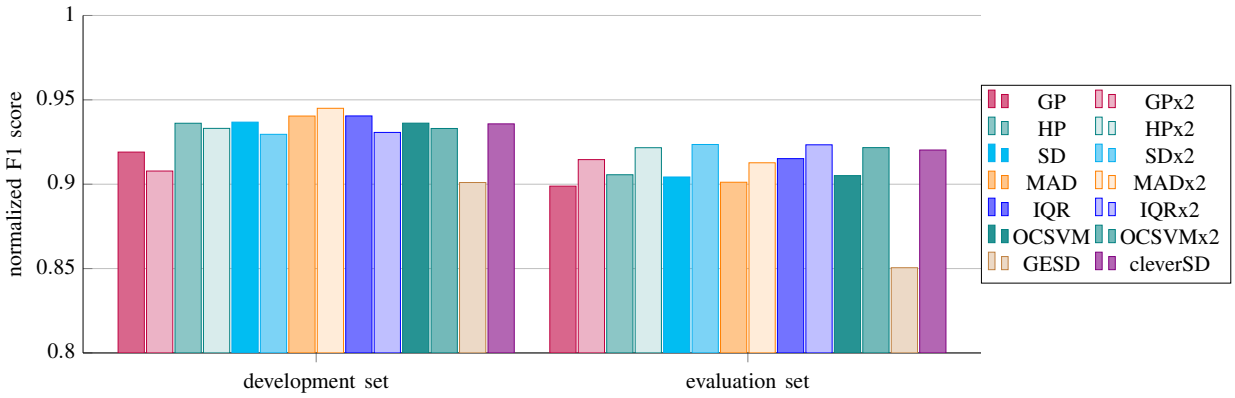

Figure 1. Comparison of decision methods based on the mean of the normalized F1 scores taken over all machine types.

all other methods. Second, the two-stage versions of the approaches, yield about the same performace on the development set and a slightly better performance on the evaluation set. Furthermore, the iterative approach cleverSD has approximately the same F1 score as the two-stage approaches. In conclusion, these experiments indicate that one should use a two-stage approach (or cleverSD) when estimating decision thresholds for ASD and the particular choice for the underlying one-stage method is not that important.

Table 3. Mean of F1 scores among all machine ids belonging to single machine types obtained with five independent trials on the evaluation dataset. Highest F1 score among different methods for each machine type is underlined.

	fan	pump	slider	ToyCar	ToyConveyor	valve	mean
GP	0.89956	0.86065	0.88863	0.60121	0.63508	0.67127	0.75940
HP	0.93845	0.89173	<u>0.92304</u>	0.58433	0.59703	0.65612	0.76512
SD	0.93675	0.89318	0.92132	0.57726	0.59974	0.65573	0.76399
MAD	0.93878	0.88953	0.91896	0.57180	0.59806	0.65103	0.76136
IQR	0.94128	<u>0.89359</u>	0.92027	0.59976	0.61941	0.66482	0.77319
OCSVM	0.93832	0.89185	0.92321	0.58277	0.59607	0.65587	0.76468
GESD	0.90994	0.88218	0.89995	0.51314	0.49565	0.61057	0.71857
cleverSD	0.94124	0.87428	0.90940	0.61826	0.64311	0.67869	0.77749
GPx2	0.90652	0.85206	0.88491	<u>0.63815</u>	<u>0.66252</u>	<u>0.69189</u>	0.77267
HPx2	0.94287	0.88472	0.91462	0.61838	0.63550	0.67583	0.77865
SDx2	0.94289	0.87921	0.91107	0.62399	0.64169	0.68273	<u>0.78026</u>
MADx2	0.94149	0.89154	0.91883	0.59180	0.61854	0.66447	0.77111
IQRx2	<u>0.94292</u>	0.87495	0.90981	0.62575	0.64314	0.68402	0.78010
OCSVMx2	0.94283	0.88521	0.91462	0.61806	0.63553	0.67594	0.77870
optimum	0.96516	0.94371	0.95913	0.75889	0.72512	0.78359	0.85593

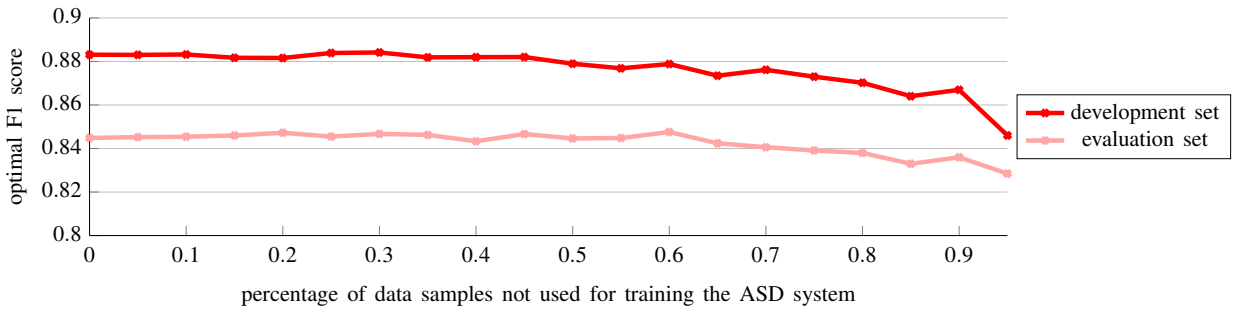

Figure 2. Mean of optimal F1 scores among machine types obtained with varying percentage of data samples not used for training the ASD system on the development set.

4.3 Dividing normal samples in disjoint sets for training ASD system and estimating decision threshold

To avoid a degraded ASD performance due to overfitting resulting from using the normal samples for estimating the decision threshold *and* training the ASD system, a commonly applied strategy is to only use a part of the normal samples for training the model and use the remaining samples for extracting more realistic scores. By using this strategy, the training scores and test scores are more similar and thus the decision threshold is expected to be more accurate. However, it is clear that using less data for training the ASD system also degrades the ASD performance since less information is incorporated into the model. In the following experiments, we investigate whether this strategy actually improves the ASD performance.

First, we evaluated the ASD performance obtained with a single optimal decision threshold for a varying number of data samples used for training the ASD system. The resulting F1 scores can be found in Figure 2. As

expected, the F1 scores decrease when using less data for training. However, the degradation in performance is much less severe than anticipated and is not noticeable before using less than 40% of normal training samples. Even when using only 5% of normal training samples, the F1 scores are only slightly worse than when using all samples. The most likely reason for this is that, ignoring the background noise, the variability of sounds emitted by machines is relatively low and thus their acoustic behavior can be captured with only a few recordings. Since this opens the possibility to train an ASD system for machine condition monitoring with much fewer computational and data resources, this observation is interesting on its own.

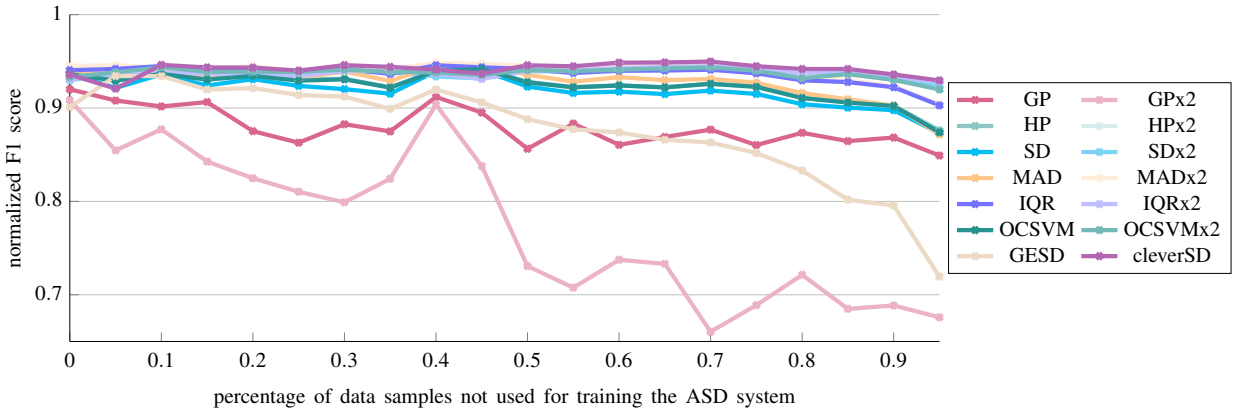

Figure 3. Mean of normalized F1 scores among machine types obtained with different methods for estimating decision thresholds and varying percentage of data samples not used for training the ASD system on the development set.

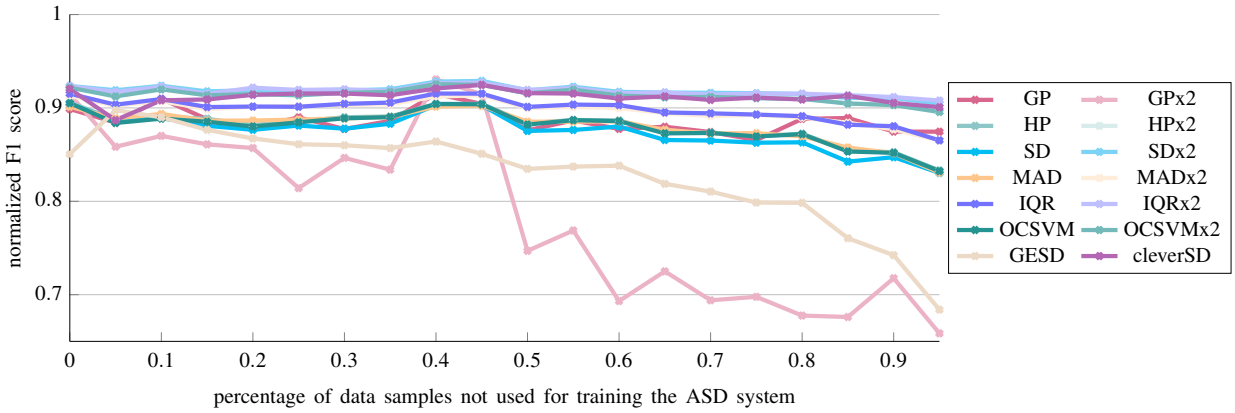

Figure 4. Mean of normalized F1 scores among machine types obtained with different methods for estimating decision thresholds and varying percentage of data samples not used for training the ASD system on the evaluation set.

Next, we evaluated the ASD performance obtained with different methods for estimating a decision threshold for a varying number of data samples used for training the ASD system. The results for the development set and evaluation set can be found in Figure 3 and Figure 4, respectively. It can be seen that using fewer samples for training the system and using these samples to estimate more realistic anomaly scores for estimating the decision threshold does not significantly improve the ASD performance. Moreover, since the absolute F1 score is actually slightly decreasing (see Figure 2) the performance is actually worse. When comparing individual methods, one can see that in general the gaps in performance get wider the less data is used for training

the ASD system. Once more, iterative approaches, namely SDx2, IQRx2, OCSVMx2 and cleverSD, perform best. Furthermore, their relative performance is relatively stable, making them a robust choice for estimating decision thresholds for ASD. Note that OCSVMx2 does not assume an underlying distribution but only linear separability of the anomaly scores. Hence, it appears to be likely that these methods, especially OCSVMx2, also work well in other settings with other ASD systems and different anomaly scores. One noticeable exception is GP, for which GPx2 the results are very noisy and much worse than every other method. Since this degraded performance is not visible to this extend when using all samples for training the ASD system, this indicates that in general the other iterative methods may be preferable.

5 CONCLUSIONS

In this work, multiple techniques for estimating decision thresholds have been reviewed and applied for detecting anomalous sounds in machine condition monitoring. In experiments conducted on the DCASE 2020 dataset, these techniques have been compared using the anomaly scores obtained with a state-of-the-art ASD system. For this particular experimental setup, the following observations have been made: First, most techniques for estimating a decision threshold perform equally well and yield approximately 90% to 95% of the F1 score obtained with an optimally tuned decision threshold. Hence, there is still a gap in performance but this gap is relatively small. Second, iterative approaches such as multi-stage thresholding [31] slightly improve the overall ASD performance and therefore are to be preferred over single-stage techniques. This is especially true when using less data for training the ASD system indicating that iterative approaches are more robust. Last but not least, holding back normal training samples (i.e. not using them for training the ASD system) for the sole purpose of obtaining more realistic anomaly scores from these samples and using the resulting scores when estimating a decision threshold does not improve the ASD performance and thus can be omitted.

Although this work is not and cannot be exhaustive in listing and comparing all methods for automatically estimating decision thresholds, it shall serve as an initial investigation on applying these techniques for practical ASD applications such as machine condition monitoring. For future work, further studies using other ASD systems for calculating the anomaly scores, other ASD datasets and additional techniques for estimating decision thresholds are to be carried out. In addition, it is planned to evaluate all mentioned methods for finding decision thresholds when dealing with domain shifts [5] and when generalizing models for multiple domains [6]. Furthermore, additional investigations on choosing decision thresholds can be led for open-set classification problems such as acoustic scene classification [46] or speaker recognition [47].

REFERENCES

[1] Aggarwal CC. Outlier Analysis. Springer; 2017.

[2] Zimek A, Filzmoser P. There and back again: Outlier detection between statistical reasoning and data mining algorithms. WIREs Data Mining Knowl Discov. 2018;8(6).

[3] Ebbers J, Haeb-Umbach R, Serizel R. Threshold Independent Evaluation of Sound Event Detection Scores. In: Proc International Conference on Acoustics, Speech and Signal Processing; 23-27 May 2022; Virtual and Singapore, China. IEEE; 2022. p. 1021-5.

[4] Koizumi Y, Kawaguchi Y, Imoto K, Nakamura T, Nikaido Y, Tanabe R, et al. Description and Discussion on DCASE2020 Challenge Task2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring. In: Proc Detection and Classification of Acoustic Scenes and Events Workshop; 2-4 November 2020; Tokyo, Japan; 2020. p. 81-5.

[5] Kawaguchi Y, Imoto K, Koizumi Y, Harada N, Niizumi D, Dohi K, et al. Description and Discussion on DCASE 2021 Challenge Task 2: Unsupervised Anomalous Detection for Machine Condition Monitoring Under Domain Shifted Conditions. In: Proc Detection and Classification of Acoustic Scenes and Events Workshop; 15-19 November 2021; Online; 2021. p. 186-90.

[6] Dohi K, Imoto K, Harada N, Niizumi D, Koizumi Y, Nishida T, et al. Description and Discussion on DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques. In arXiv e-prints: 220605876. 2022.

[7] Bai J, Wang M, Chen J. Dual-Path Transformer For Machine Condition Monitoring. In: Proc Asia-Pacific Signal and Information Processing Association Annual Summit and Conference; 14-17 December 2021; Tokyo, Japan. IEEE; 2021. p. 1144-8.

[8] Li R, Gu X, Lu F, Song H, Pan J. Unsupervised Adversarial domain adaptive abnormal sound detection for machine condition monitoring under Domain Shift Conditions. DCASE2021 Challenge; Tech Rep; 2021.

[9] Narita H, Tamamori A. Unsupervised Anomalous Sound Detection Using Intermediate Representation of Trained Models and Metric Learning Based Variational Autoencoder. DCASE2021 Challenge; Tech Rep; 2021.

[10] Pham L, Jalali A, Dinica O, Schindler A. DCASE Challenge 2021: Unsupervised Anomalous Sound Detection of Machinery with LeNet Architecture. DCASE2021 Challenge; Tech Rep; 2021.

[11] Wilkinghoff K. Combining Multiple Distributions based on Sub-Cluster AdaCos for Anomalous Sound Detection under Domain Shifted Conditions. In: Proc Detection and Classification of Acoustic Scenes and Events Workshop; 15-19 November 2021; Online; 2021. p. 55-9.

[12] Zhang C, Yao Y, Qiu R, Li S, Shao X. Unsupervised Anomalous Sound Detection Using Denoising-Detection System Under Domain Shifted Conditions. DCASE2021 Challenge; Tech Rep; 2021.

[13] Kapka S. ID-Conditioned Auto-Encoder for Unsupervised Anomaly Detection. In: Proc 5th Workshop on Detection and Classification of Acoustic Scenes and Events; 2-4 November 2020; Tokyo, Japan (full virtual); 2020. p. 71-5.

[14] Giri R, Cheng F, Helwani K, Tenneti SV, Isik U, Krishnaswamy A. Group Masked Autoencoder Based Density Estimator for Audio Anomaly Detection. In: Proc 5th Workshop on Detection and Classification of Acoustic Scenes and Events; 2-4 November 2020; Tokyo, Japan (full virtual); 2020. p. 51-5.

[15] Hendrycks D, Mazeika M, Dietterich TG. Deep Anomaly Detection with Outlier Exposure. In: Proc 7th International Conference on Learning Representations; 6-9 May 2019; New Orleans, LA, USA. OpenReview.net; 2019. p. 1-18.

[16] Deng J, Guo J, Xue N, Zafeiriou S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In: Conference on Computer Vision and Pattern Recognition; 16-20 June 2019; Long Beach, CA, USA. IEEE; 2019. p. 4690-9.

[17] Zhang X, Zhao R, Qiao Y, Wang X, Li H. AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations. In: Proc Conference on Computer Vision and Pattern Recognition; 16-20 June 2019; Long Beach, CA, USA. IEEE; 2019. p. 10823-32.

[18] Zhou Q. ArcFace Based Sound Mobilenets for DCASE 2020 task 2. DCASE2020 Challenge; Tech Rep; 2020.

[19] Lopez JA, Lu H, Lopez-Meyer P, Nachman L, Stemmer G, Huang J. A Speaker Recognition Approach to Anomaly Detection. In: Proc 5th Workshop on Detection and Classification of Acoustic Scenes and Events; 2-4 November 2020; Tokyo, Japan (full virtual); 2020. p. 96-9.

[20] Lopez JA, Stemmer G, Lopez-Meyer P, Singh P, del Hoyo Ontiveros JA, Cordourier HA. Ensemble Of Complementary Anomaly Detectors Under Domain Shifted Conditions. In: Proc Detection and Classification of Acoustic Scenes and Events Workshop; 15-19 November 2021; Online; 2021. p. 11-5.

[21] Kuroyanagi I, Hayashi T, Adachi Y, Yoshimura T, Takeda K, Toda T. An Ensemble Approach to Anomalous Sound Detection Based on Conformer-Based Autoencoder and Binary Classifier Incorporated with Metric Learning. In: Proc Detection and Classification of Acoustic Scenes and Events Workshop; 15-19 November 2021; Online; 2021. p. 110-4.

[22] Sakamoto Y, Miyamoto N. Combine Mahalanobis Distance, Interpolation Auto Encoder and Classification Approach for Anomaly Detection. DCASE2021 Challenge; Tech Rep; 2021.

[23] Morita K, Yano T, Tran K. Anomalous Sound Detection Using CNN-Based Features By Self Supervised Learning. DCASE2021 Challenge; Tech Rep; 2021.

[24] Wilkinghoff K. Sub-Cluster AdaCos: Learning Representations for Anomalous Sound Detection. In: Proc International Joint Conference on Neural Networks; 18-22 July 2021; Shenzhen, China. IEEE; 2021. p. 1-8.

[25] He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: Proc Conference on Computer Vision and Pattern Recognition; 27-30 June 27-30 2016; Las Vegas, NV, USA. IEEE; 2016. p. 770-8.

[26] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, et al. Tensorflow: A system for large-scale machine learning. In: Proc 12th USENIX Symposium on Operating Systems Design and Implementation; 2-4 November 2016; Savannah, GA, USA; 2016. p. 265-83.

[27] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. In: Proc 3rd International Conference on Learning Representations; 7-9 May 2015; San Diego, CA, USA; 2015. p. 1-15.

[28] Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. Mixup: Beyond empirical risk minimization. In: Proc International Conference on Learning Representations; 30 April - 3 May 2018; Vancouver, BC, Canada; 2018. p. 1-13.

[29] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011;12:2825-30.

[30] Reimann C, Filzmoser P, Garrett RG. Background and threshold: critical comparison of methods of determination. Science of the total environment. 2005;346(1-3):1-16.

[31] Yang J, Rahardja S, Fränti P. Outlier detection: How to threshold outlier scores? In: Proc International Conference on Artificial Intelligence, Information Processing and Cloud Computing; 19-21 December 2019; Sanya, China. ACM; 2019. p. 37:1-37:6.

[32] Gao J, Tan P. Converting Output Scores from Outlier Detection Algorithms into Probability Estimates. In: Proc 6th International Conference on Data Mining; 18-22 December 2006; Hong Kong, China. IEEE; 2006. p. 212-21.

[33] Gebel M. Multivariate calibration of classifier scores into the probability space [Ph.D. thesis]. University of Dortmund; 2009.

[34] Rousseeuw PJ, Van Zomeren BC. Unmasking multivariate outliers and leverage points. Journal of the American Statistical association. 1990;85(411):633-9.

[35] Filzmoser P. A multivariate outlier detection method. In: Proc 7th International Conference on Computer Data Analysis and Modeling; Minsk, Belarus; 2004. p. 18-22.

[36] Clark J, Liu Z, Japkowicz N. Adaptive Threshold for Outlier Detection on Data Streams. In: Proc 5th International Conference on Data Science and Advanced Analytics; 1-3 October 2018; Turin, Italy. IEEE; 2018. p. 41-9.

[37] Gökcesu K, Neyshabouri MM, Gökcesu H, Kozat SS. Sequential Outlier Detection Based on Incremental Decision Trees. IEEE Trans Signal Process. 2019;67(4):993-1005.

[38] Zhang M, Li X, Wang L. An Adaptive Outlier Detection and Processing Approach Towards Time Series Sensor Data. IEEE Access. 2019;7:175192-212.

[39] Rousseeuw PJ, Croux C. Alternatives to the median absolute deviation. Journal of the American Statistical association. 1993;88(424):1273-83.

[40] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the Support of a High-Dimensional Distribution. Neural Comput. 2001;13(7):1443-71.

[41] Rosner B. Percentage Points for a Generalized ESD Many-Outlier Procedure. Technometrics. 1983;25(2):165-72.

[42] Grubbs FE. Sample Criteria for Testing Outlying Observations. The Annals of Mathematical Statistics. 1950;21(1):27 58.

[43] Buzzi-Ferraris G, Manenti F. Outlier detection in large data sets. Computers & chemical engineering. 2011;35(2):388-90.

[44] Koizumi Y, Saito S, Uematsu H, Harada N, Imoto K. ToyADMOS: A Dataset of Miniature-Machine Operating Sounds for Anomalous Sound Detection. In: Proc Workshop on Applications of Signal Processing to Audio and Acoustics; October 20-23, 2019; New Paltz, NY, USA. IEEE; 2019. p. 313-7.

[45] Purohit H, Tanabe R, Ichige T, Endo T, Nikaido Y, Suefusa K, et al. MIMII Dataset: Sound Dataset for Malfunctioning Industrial Machine Investigation and Inspection. In: Proc Detection and Classification of Acoustic Scenes and Events Workshop; 25-26 October 2019; New York, NY, USA; 2019. p. 209-13.

[46] Mesaros A, Heittola T, Virtanen T. Acoustic Scene Classification in DCASE 2019 Challenge: Closed and Open Set Classification and Data Mismatch Setups. In: Proc Workshop on Detection and Classification of Acoustic Scenes and Events; 25-26 October 2019; New York, USA; 2019. p. 164-8.

[47] Shon S, Dehak N, Reynolds DA, Glass JR. MCE 2018: The 1st Multi-Target Speaker Detection and Identification Challenge Evaluation. In: Proc 20th Annual Conference of the International Speech Communication Association; 15-19 September 2019; Graz, Austria. ISCA; 2019. p. 356-60.