AN OUTLIER EXPOSED ANOMALOUS SOUND DETECTION SYSTEM FOR DOMAIN GENERALIZATION IN MACHINE CONDITION MONITORING

Technical Report

Kevin Wilkinghoff

Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE
Fraunhoferstraße 20, 53343 Wachtberg, Germany
kevin.wilkinghoff@fkie.fraunhofer.de

ABSTRACT

Emitted machine sounds can change drastically due to a change in settings of machines or due to varying noise conditions. This is a problem when monitoring the condition of these machines with a trained anomalous sound detection system because after changing the acoustic conditions the normal sounds are often falsely marked as anomalous. The goal of task 2 "Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques" of the DCASE 2022 challenge is to develop systems that reliably detect anomalous sounds regardless of whether characteristics of machine sounds are changing or not. In this work, a conceptually simple outlier exposed anomalous sound detection system is presented that is specifically designed for domain generalization. To this end, multiple feature representations and carefully designed sub-system architectures are utilized inside a single neural network. Furthermore, a technique called domain mixup is presented to further improve the domain generalization capabilities.

Index Terms— anomalous sound detection, domain generalization, domain shift, machine listening

1. INTRODUCTION

The goal of anomalous sound detection (ASD) is to recognize sounds substantially differing from normal sounds that are frequently encountered. Depending on the particular application, defining which sounds exactly should be considered *normal* can be challenging. In the context of machine condition monitoring, defining the normal class is relatively simple: Sounds emitted by machines that are properly working are normal, any deviations from the intended behaviour indicate mechanical failure and resulting sounds should be considered anomalous.

Research of ASD for machine condition monitoring is heavily promoted through tasks of the annual DCASE challenge [1, 2]. These ASD tasks are in a semi-supervised setting, meaning that only normal data is available for training the system. In 2021, the ASD task focused on ASD in domain-shifted conditions. This means that there are two data domains differing in acoustic conditions or machine attributes as for example speed or size: a source domain with many training samples and a target domain with only very few training samples. The goal is to adapt an ASD system trained on data belonging to the source domain to also work in the target domain. This prevents the need to collect new recordings of machines and retraining an ASD system when changing machine settings or noise conditions. The ASD system described in this work is designed for task 2 of the DCASE 2022 challenge, titled "Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques" [3]. The major difference between domain generalization [4] and domain-shift is that an ASD system should correctly detect anomalies regardless of whether sounds belong to the source or any target domain. Thus, when detecting anomalous sounds knowledge about the domain is not available and the same model and decision threshold for detecting anomalies must be used for source and target domains.

The dataset of the DCASE 2022 challenge task 2 consists of sounds from the machine types "bearing", "fan", "gearbox", "slider", "valve" from MIMII DG [5], and "ToyCar", "ToyTrain" from ToyADMOS2 [6]. For each machine type, there are 6 different subsets of the dataset called *sections*, of which three belong to a development set and the other three belong to an evaluation set, corresponding to different types of domain shifts. For each section, there are 990 normal training samples belonging to the source domain, 10 normal training samples belonging to a target domain and 200 test samples each belonging to one of the domains. Furthermore, some attribute information are given for normal training samples defining states of the machines or different types of noise. All recordings include real factory noise, have a length of 10 seconds and a sampling rate of 16 kHz.

There are two main state-of-the-art strategies to train an ASD system. Both are realized as baseline systems by the organizers of the challenge. First, an autoencoder can be trained with normal data belonging to a single machine only. Assuming that anomalous data substantially differs from normal data and thus cannot be reconstructed as well as normal data, the reconstruction error can then be used as an anomaly score. Second, a model can be trained to discriminate among different machine types or among other acoustic characteristics such as different machine settings or noise conditions. Here, the assumption is that the information needed for correctly classifying the sounds is also sufficient to detect anomalous sounds. This is called an outlier exposed ASD system [7].

The main contribution of this work is to present an outlier exposed anomalous sound detection system with strong domain generalization capabilities in machine condition monitoring [1]. The system is conceptually simple since its architecture and hyperparameter settings are the same for each machine type. Furthermore, no external data resources have been used to train the system or augment the data. The most important and novel design choices of the sys-

[1] An open-source implementation of the proposed system is available at: `https://github.com/wilkinghoff/dcase2022`

tem are the following: First, multiple input feature representations, namely log-mel spectrograms as well as magnitude spectra are used. Second, the network architectures are carefully designed to avoid learning trivial mappings to hyperspheres and thus less meaningful embeddings. Last but not least, a simple technique called domain mixup for improving the domain generalization capabilities when estimating the distribution of the embeddings is proposed.

2. PROPOSED SYSTEM

The overall architecture of the proposed system is shown in Fig. 1. All details of the individual blocks can be found in the following subsections.

2.1. Data preprocessing

The system utilizes two different feature representations derived from the raw waveforms. First, log-mel spectrograms with a Hanning-windowed DFT length of 1024, a hop size of 256 and 128 mel bins are used as done in many state-of-the-art ASD systems [8, 9, 10, 11, 12]. Second, magnitude spectra of the entire signals, denoted by *DFT representations*, are used to better capture the characteristics of machines that emit a relatively stationary sound as for example fans. To reduce acoustic differences between source and target domains, sample-wise temporal mean normalization similar to cepstral mean normalization (CMN) [13] is applied to the log-mel spectrograms.

2.2. Neural network architecture

The neural network for extracting the embeddings consists of two different sub-networks for each input representation and is trained to jointly discriminate among the machine types, sections and different attribute information about the machines resulting in a total of 342 classes. In contrast to [9], where multiple networks or loss functions have been used for different classification tasks, this training strategy is much simpler. Thus, all information given for the normal training data needs to be captured inside the embeddings. Since both input representations have entirely different dimensions, two different neural network architectures are used for further processing. The sub-network used for the log-mel spectrograms is based on a modified ResNet architecture [14] and is described in Tab. 1. The same architecture has been successfully applied for ASD with a sub-cluster AdaCos loss in [9] and for ASD with domain adaptation [10]. For the DFT representations, the sub-network consists of three one-dimensional convolutions and five dense layers as shown in Tab. 2.

Both sub-network architectures have been carefully designed to avoid learning trivial mappings to hyperspheres for specific classes as done in networks for deep one-class classification [15]. This means that 1) no bounded non-linearities, 2) no bias terms and 3) no trainable hypersphere centers are used. Instead, for 1) we only leaky rectified linear units (LeakyReLU) with $\alpha = 0.1$ [16] as non-linearities and for 3) we randomly initialize the cluster centers of the sub-cluster AdaCos loss without adapting them during training. A random initialization of the cluster centers is not a problem, since the embeddings and the cluster centers live in a relatively high-dimensional space (256 dimensions) and thus are very likely to be pairwise orthogonal. Furthermore, no batch normalization (BN) [17] is applied inside the convolutional blocks but only before the blocks or after the flattening layer.

Table 1: Modified ResNet architecture for log-mel spectrograms.

layer name	structure	output size
input	BN (temporal axis)	622×128
2D convolution	7×7, stride$= 2$	$311 \times 64 \times 16$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$155 \times 31 \times 16$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$78 \times 16 \times 32$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$39 \times 8 \times 64$
residual block	$\begin{pmatrix} 3 \times 3 \\ 3 \times 3 \end{pmatrix} \times 2$, stride$= 1$	$20 \times 4 \times 128$
max pooling	20×1, stride$= 1$	$1 \times 4 \times 128$
flatten	BN	512
dense (embedding)	linear	128

Table 2: Network architecture for DFT representations.

layer name	structure	output size
input	-	80000
1D convolution	256, stride$= 64$	1250×128
1D convolution	64, stride$= 16$	40×128
1D convolution	16, stride$= 4$	10×128
flatten	-	1280
dense	BN, Leaky ReLU	128
dense	BN, Leaky ReLU	128
dense	BN, Leaky ReLU	128
dense	BN, Leaky ReLU	128
dense (embedding)	linear	128

The output of both sub-networks, which can both be interpreted as embeddings by themselves, are concatenated in order to obtain a single embedding for each file. This concatenation ensures that *both* networks capture all information needed to discriminate among the classes present in their respective feature representations. Therefore, the embeddings are more sensitive to anomalous sounds than when giving the network the freedom to utilize only a single feature representation (e.g. by taking the sum) because specific anomalies may be clearly apparent in only one of the two input representations.

Most outlier exposed ASD systems utilize angular margin losses such as ArcFace [18] or AdaCos [19], to train a network for extracting embeddings. AdaCos is based on an adaptive scale parameter and does not require any tuning of hyperparameters. Here, sub-cluster AdaCos [9] with 16 sub-clusters, which is an extension of AdaCos [19] specifically designed for ASD tasks, has been used. The main idea of the sub-cluster AdaCos loss is to use multiple clusters per class instead of a single one in order to learn more complex distributions for the resulting embeddings. Embeddings obtained with this loss have been shown to outperform embeddings obtained with regular AdaCos for ASD tasks [9]. The entire network is implemented using Tensorflow [20] and is trained for 400 epochs with a batch size of 64 using Adam [21]. For data augmentation, only mixup [22] with a uniformly distributed mixing coefficient is used.

2.3. Calculating anomaly scores

In [9], it has been shown that using Gaussian mixture models (GMMs) with a full covariance matrix for estimating the distribution of the embeddings and calculating anomaly scores results in a better ASD performance than taking other backends such as cosine similarity. Hence, GMMs with 16 Gaussian components and a regularized covariance matrix by adding 10^{-3} to the diagonal as implemented in scikit-learn [23] are trained and their resulting log-likelihood values are used as anomaly scores. This strategy has

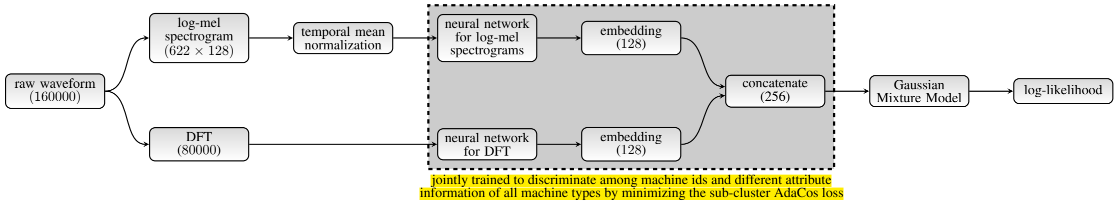

Figure 1: Structure of the proposed anomalous sound detection system.

also been applied in domain-shifted conditions by estimating multiple distributions for combinations of machine types and sections, source and target domains, and attribute information [10]. Since the system proposed in this work is specifically designed for domain generalization and thus domain knowledge is not available during testing, only one GMM is trained for each combination of machine type and section resulting in a total of 21 GMMs for the development and 21 GMMs for the evaluation set. Note, that it is also possible to estimate whether a sample belongs to the source or target domain and train different models for each domain but this would contradict the goal of domain generalization and is the reason why we did not employ such a strategy.

To further improve the domain generalization capabilities, we propose a technique called *domain mixup* when training the GMMs. For each combination of machine type and section, every embedding corresponding to a normal training sample of the source domain is mixed with a random sample of the target domain by taking the mean of both embeddings. Therefore, essentially there is an additional copy of each sample belonging to the source domain consisting of a mixed-up sample with the target domain. Then, each GMM is trained with these mixed-up samples as well as the original samples from the source domain. This procedure is a simple way to generate additional training samples belonging to the correct machine type and section but neither belonging to the source nor to the target domain. Thus, by using these newly generated samples a distribution of the embeddings that is more independent of the domain can be learned.

2.4. Ensembling strategy

To improve the performance of the system, a similar ensembling strategy as used in [10] has been applied. More concretely, the proposed system is trained ten times and after every 100 of the 400 training epochs the model is stored. Thus, in total $10 \times 4 = 40$ different models and corresponding sets of embeddings are obtained. In contrast to [10], the same number of sub-clusters in the sub-cluster AdaCos loss is used, namely 16, when retraining the system. For each set of embeddings, another GMM is trained using all embeddings belonging to a single section and machine type. The sum of the resulting log-likelihoods is taken to obtain a single ASD score for this combination of section and machine type.

2.5. Setting decision thresholds

For setting decision thresholds, the 90th percentile of the anomaly scores of all normal training samples belonging to a given section is calculated. All anomaly scores of test samples belonging to the same section that are above this threshold are marked as anomalous. This is the same strategy as applied in [10].

2.6. Submissions

For the challenge, two slightly different systems have been submitted. The first system is the proposed system as described before. As a small deviation from this, the same system has been submitted without applying domain mixup when calculating the anomaly scores in order to investigate the impact of domain mixup on the ASD performance.

3. RESULTS

The ASD results obtained on the development set with the proposed system compared to both baseline systems can be found in Tab. 3. It is clearly visible that the proposed system significantly outperforms both baseline systems in terms of AUC and pAUC scores regardless of machine type and domain. Since random guessing corresponds to an AUC score of 50%, both baseline systems fail miserably when predicting anomalous machine sounds for some target domains. One example is the target domain of section 2 belonging to the machine type “ToyTrain” where the autoencoder baseline system has an AUC of less than 15% and the MobileNetV2-based system as an AUC score of less than 45%. In contrast to that, the proposed system in many cases has a performance close to the one obtained in the source domain and always performs at least as well as random guessing. However, in some cases there is still a substantial performance gap e.g. for the machine type “ToyTrain”, section 0 or 1. Still, the domain generalization capabilities of the proposed system, which is the main focus of this challenge, are far superior to those of the baseline systems.

4. CONCLUSIONS

In this work, a conceptually simple outlier exposed anomalous sound detection system with strong domain generalization capabilities submitted to the DCASE challenge 2022 has been presented. The system is based on a neural network trained with the sub-cluster AdaCos loss to extract discriminative embeddings and consists of two carefully designed sub-systems utilizing log-mel spectrograms and magnitude spectra as input feature representations. To improve the domain generalization capabilities, a simple technique called domain mixup, which mixes samples of the source and target domain when estimating the distribution of the embeddings with a GMM to obtain anomaly scores, has been presented. No external data has been used to augment data samples or train the system. In experiments conducted on the development dataset belonging to task 2 “Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques” of the DCASE 2022 challenge, it has been shown that the proposed system significantly outperforms both baseline systems on source as well as target domains for each machine type.

5. REFERENCES

[1] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on DCASE2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 81–85.

[2] Y. Kawaguchi, K. Imoto, Y. Koizumi, N. Harada, D. Niizumi, K. Dohi, R. Tanabe, H. Purohit, and T. Endo, "Description and discussion on dcase 2021 challenge task 2: Unsupervised anomalous detection for machine condition monitoring under domain shifted conditions," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 186–190.

[3] K. Dohi, K. Imoto, N. Harada, D. Niizumi, Y. Koizumi, T. Nishida, H. Purohit, T. Endo, M. Yamamoto, and Y. Kawaguchi, "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: 2206.05876*, 2022.

[4] J. Wang, C. Lan, C. Liu, Y. Ouyang, and T. Qin, "Generalizing to unseen domains: A survey on domain generalization," in *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*. ijcai.org, 2021, pp. 4627–4635.

[5] K. Dohi, T. Nishida, H. Purohit, R. Tanabe, T. Endo, M. Yamamoto, Y. Nikaido, and Y. Kawaguchi, "MIMII DG: Sound dataset for malfunctioning industrial machine investigation and inspection for domain generalization task," *In arXiv e-prints: 2205.13879*, 2022.

[6] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, and S. Saito, "ToyADMOS2: Another dataset of miniature-machine operating sounds for anomalous sound detection under domain shift conditions," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, November 2021, pp. 1–5.

[7] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *7th International Conference on Learning Representations (ICLR)*. OpenReview.net, 2019.

[8] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 46–50.

[9] K. Wilkinghoff, "Sub-cluster AdaCos: Learning representations for anomalous sound detection," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021.

[10] ——, "Combining multiple distributions based on sub-cluster AdaCos for anomalous sound detection under domain shifted conditions," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 55–59.

[11] J. A. Lopez, G. Stemmer, P. Lopez-Meyer, P. Singh, J. A. del Hoyo Ontiveros, and H. A. Cordourier, "Ensemble of complementary anomaly detectors under domain shifted conditions," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 11–15.

[12] I. Kuroyanagi, T. Hayashi, Y. Adachi, T. Yoshimura, K. Takeda, and T. Toda, "An ensemble approach to anomalous sound detection based on conformer-based autoencoder and binary classifier incorporated with metric learning," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2021, pp. 110–114.

[13] A. E. Rosenberg, C. Lee, and F. K. Soong, "Cepstral channel normalization techniques for hmm-based speaker verification," in *ICSLP*. ISCA, 1994.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 770–778.

[15] L. Ruff, N. Görnitz, L. Deecke, S. A. Siddiqui, R. A. Vandermeulen, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *35th International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 4390–4399.

[16] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *30th International Conference on Machine Learning (ICML)*, 2013.

[17] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning (ICML)*, vol. 37, 2015, pp. 448–456.

[18] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 4690–4699.

[19] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, "AdaCos: Adaptively scaling cosine logits for effectively learning deep face representations," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 10 823–10 832.

[20] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016, pp. 265–283.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, Y. Bengio and Y. LeCun, Eds., 2015.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations (ICLR)*, 2018.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Table 3: AUCs and pAUCs per machine type obtained with both baseline systems and the proposed system. The last row for each machine type contains the harmonic mean of the values for all three sections obtained with a single threshold for source and target domain. Highest AUCs and pAUCs in each row are underlined.

dataset split			baselines				proposed system	
			autoencoder		MobileNetV2			
machine type	section	domain	AUC	pAUC	AUC	pAUC	AUC	pAUC
ToyCar	0	source	84.64%	68.00%	48.50%	49.68%	83.52%	72.42%
ToyCar	0	target	39.50%	49.89%	49.64%	48.74%	74.00%	56.84%
ToyCar	1	source	86.04%	74.74%	58.36%	51.26%	80.00%	52.42%
ToyCar	1	target	38.64%	51.47%	58.62%	49.05%	59.96%	53.05%
ToyCar	2	source	98.66%	94.74%	72.74%	81.05%	99.64%	98.32%
ToyCar	2	target	32.56%	52.32%	43.88%	49.05%	88.32%	67.37%
ToyCar	harmonic mean	mixed	63.30%	53.09%	54.92%	49.85%	81.16%	63.99%
ToyTrain	0	source	68.58%	71.79%	51.40%	47.79%	69.88%	61.89%
ToyTrain	0	target	31.38%	49.68%	40.14%	52.00%	54.60%	49.89%
ToyTrain	1	source	80.78%	69.05%	76.18%	70.11%	92.64%	83.79%
ToyTrain	1	target	29.46%	49.89%	45.82%	50.00%	54.08%	50.11%
ToyTrain	2	source	84.66%	78.53%	70.00%	55.16%	99.92%	99.58%
ToyTrain	2	target	14.28%	48.21%	39.42%	53.47%	87.32%	73.05%
ToyTrain	harmonic mean	mixed	51.40%	50.39%	53.08%	51.18%	70.22%	55.77%
bearing	0	source	52.56%	56.74%	77.60%	62.11%	61.56%	62.53%
bearing	0	target	63.28%	49.79%	73.24%	64.74%	63.56%	53.05%
bearing	1	source	75.22%	65.79%	70.88%	64.11%	78.12%	64.00%
bearing	1	target	62.60%	55.58%	70.38%	61.05%	88.92%	80.00%
bearing	2	source	39.42%	47.89%	71.18%	53.89%	84.76%	71.58%
bearing	2	target	46.20%	50.74%	52.90%	50.63%	76.04%	68.84%
bearing	harmonic mean	mixed	54.41%	51.96%	68.91%	58.32%	73.88%	63.38%
fan	0	source	84.04%	81.37%	76.08%	65.26%	99.84%	99.16%
fan	0	target	34.90%	60.42%	50.76%	51.16%	93.56%	82.11%
fan	1	source	72.20%	54.00%	67.26%	49.68%	95.04%	92.00%
fan	1	target	44.98%	51.68%	39.90%	50.63%	71.64%	51.58%
fan	2	source	78.74%	73.16%	81.88%	73.89%	88.04%	83.79%
fan	2	target	64.60%	60.21%	68.36%	64.42%	86.64%	72.42%
fan	harmonic mean	mixed	62.71%	58.36%	62.84%	55.42%	90.66%	78.88%
gearbox	0	source	64.02%	64.42%	48.22%	56.32%	87.88%	78.11%
gearbox	0	target	64.08%	60.84%	52.26%	56.84%	82.88%	74.11%
gearbox	1	source	68.14%	54.63%	51.26%	54.53%	90.64%	72.21%
gearbox	1	target	57.74%	53.16%	56.64%	56.73%	83.96%	54.53%
gearbox	2	source	75.24%	66.00%	77.20%	75.05%	86.68%	78.11%
gearbox	2	target	65.96%	60.53%	36.78%	49.47%	88.84%	57.47%
gearbox	harmonic mean	mixed	65.70%	59.32%	53.58%	53.99%	85.39%	68.64%
slide rail	0	source	80.88%	71.26%	94.32%	83.26%	98.52%	96.00%
slide rail	0	target	56.50%	54.95%	84.16%	64.95%	91.12%	65.26%
slide rail	1	source	67.96%	52.00%	45.04%	48.21%	99.60%	97.89%
slide rail	1	target	49.58%	53.37%	22.67%	47.37%	93.40%	85.89%
slide rail	2	source	87.28%	66.95%	85.50%	72.95%	84.44%	82.95%
slide rail	2	target	38.60%	52.84%	24.20%	47.89%	78.84%	69.05%
slide rail	harmonic mean	mixed	63.21%	56.09%	50.88%	53.96%	89.06%	75.81%
valve	0	source	54.40%	54.84%	64.22%	62.53%	99.72%	98.53%
valve	0	target	52.20%	50.42%	41.28%	53.47%	75.92%	64.63%
valve	1	source	50.28%	49.37%	59.54%	56.53%	86.84%	77.05%
valve	1	target	51.54%	50.32%	67.14%	56.53%	100.00%	100.00%
valve	2	source	51.50%	48.74%	77.16%	85.79%	99.04%	94.95%
valve	2	target	43.62%	49.79%	77.40%	85.79%	66.12%	71.37%
valve	harmonic mean	mixed	50.48%	50.29%	62.91%	63.44%	90.00%	80.34%
all	harmonic mean	mixed	58.13%	54.00%	57.52%	54.86%	82.18%	68.48%