UTILIZING SUB-CLUSTER ADACOS FOR ANOMALOUS SOUND DETECTION UNDER DOMAIN SHIFTED CONDITIONS

Technical Report

Kevin Wilkinghoff

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

ABSTRACT

Anomalous sound detection systems based on sub-cluster AdaCos yield state-of-the-art performance on the DCASE 2020 dataset for anomalous sound detection. In contrast to the previous year, the dataset belonging to task 2 "Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions" of the DCASE challenge 2021 contains not only source domains with 1000 normal training samples for each machine but also so-called target domains with different acoustic conditions for which only 3 normal training samples are available. To address this additional problem, a novel anomalous sound detection system based on sub-cluster AdaCos for the DCASE challenge 2021 is presented. This system is trained to extract embeddings whose distributions are estimated in different ways for source and target domains, and utilize their negative log-likelihoods as anomaly scores. In experimental evaluations, it is shown that the presented system significantly outperforms both baseline systems on source and target domains of the development set.

Index Terms— anomalous sound detection, machine listening, representation learning, angular margin loss, domain shift

1. INTRODUCTION

The goal of semi-supervised anomalous sound detection is to decide whether a given audio sample resembles the training data i.e. is *normal* or substantially differs from the training data and thus is *anomalous*. Basically, one can distinguish two major strategies for anomalous sound detection: The first approach is based on training autoencoders to encode normal data into a lower-dimensional space and then reconstruct it again [1, 2]. The underlying assumption is that normal data can be reconstructed well after training while anomalous data cannot leading to a higher reconstruction error. Thus, the reconstruction error can be used as an anomaly score. The second approach is to train neural networks to discriminate among classes as for example machine types and utilize the trained neural network to extract representation of the data, so-called embeddings, as features [3, 4, 5, 6, 7]. Here, the assumption is that the information needed to discriminate among the classes and thus is contained in the embeddings is also sufficient to distinguish normal from anomalous data. Angular margin losses such as ArcFace [8] or AdaCos [9], which ensure a margin between the classes, have been shown to outperform standard softmax losses in this context. To our knowledge, the best performing system on the anomalous sound detection dataset belonging to task 2 of the DCASE challenge 2020 [10] uses an extension of AdaCos, called sub-cluster AdaCos [11]. This loss learns more than a single mean value for each class to estimate less restrictive distributions of the embeddings than standard AdaCos and utilizes Gaussian mixture models (GMMs) to estimate these distributions for the normal data instead of comparing embeddings to the learned mean values by using the cosine similarity. This superior performance is the reason why this work focuses entirely on a system based on the sub-cluster AdaCos loss.

The system presented in this paper is designed for and submitted to task 2 "Unsupervised Anomalous Sound Detection for Machine Condition Monitoring under Domain Shifted Conditions" of the DCASE challenge 2021 [12]. The dataset of this task consists of audio recordings with a length of 10 seconds and a sampling rate of 16 kHz belonging to the machine types "ToyCar" and "ToyTrain" from ToyADMOS2 [13] and the machines types "fan", "gearbox", "pump", "slide rail" and "valve" from MIMII DUE [14]. The organizers of the challenge also provided two baseline systems: An autoencoder, which is the same as the baseline system of the previous edition of the task, and a discriminatively trained MobileNetV2-based baseline that predicts the section a given audio sample belongs to. Both baseline systems have a similar overall performance when detecting anomalous data.

In contrast to the DCASE challenge 2020, there are several differences for this year's task: First and foremost, the dataset is split into source domains for which about 1000 normal training samples are provided for each of the 6 sections per machine type and so-called target domains for the same sections with different acoustic conditions than the source domains for which only 3 normal training samples are available. For both domains, the same number of test samples is provided, about 100 normal samples and 100 anomalous samples. Furthermore, the dataset is split into a development set consisting of half of the sections and an evaluation set consisting of the other half of the sections. Another difference between the datasets is, that the sections to not directly correspond to specific products of a machine type but the same products can appear in different sections or different products can appear in the same sections. Both of these changes make the task much more challenging than before. Last but not least, the DCASE 2020 dataset consists of slightly different machine types, namely "ToyCar" and "ToyConveyor" from the ToyADMOS dataset [15] and the machine types "fan", "pump", "slide rail" and "valve" from the MIMII dataset [16].

The goal of this work is to investigate how to utilize the sub-cluster AdaCos loss for the DCASE 2021 anomalous sound detection dataset with its novel challenges. To this end, a system based

Table 1: Modified ResNet architecture used for all experiments.

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	-	42
sub-cluster AdaCos	-	199

on the sub-cluster AdaCos loss is presented. As a second contribution, different ways to compute anomaly scores for the source and target domains are proposed. Furthermore, it is shown how to decide whether samples are normal or anomalous only based on these scores from normal data. In experimental evaluations, it is shown that the proposed system significantly outperforms both baseline systems on the source and target domains of the development set.

2. PROPOSED METHOD

2.1. Data preprocessing

To compute input features for the neural network, log-Mel spectrograms with 128 Mel-bins, a window size of 1024 and a hop size of 512 are extracted from all raw waveforms with a sampling rate of 16 kHz resulting in features of size 313×128. These features are then standardized by subtracting the temporal mean and dividing by the temporal standard deviation estimated from all training files.

2.2. Neural network architecture

The network architecture used throughout this work is the same as used in [11] and can be found in Tab. 1. It consists of several residual blocks [17] whose output is further processed by max-pooling over time, flattening and a linear dense layer to obtain the embeddings of size 128. In each residual block, batch normalization [18] is applied and LeakyReLu [19] with $\alpha = 0.1$ is used as the non-linear transfer function.

To train the neural network, two sub-cluster AdaCos losses [11] with equal weight are minimized using Adam [20]. One is for classifying jointly among the sections and machine types and the other one for classifying among the different attribute information given in the filenames. When training, all normal data contained in the training set and the additional training set has been used resulting in a total of 42 sections and 199 different attribute information. Furthermore, mixup [21] is used during training to avoid overfitting of the model to the training data. The network is implemented in Tensorflow [22] and trained for 400 epochs with a batch size of 64.

2.3. Calculating anomaly scores

Throughout this work, all anomaly scores are computed by training Gaussian mixture models (GMMs) on the embeddings and utilizing negative weighted log-likelihoods as scores. In [11], it has been shown that using GMMs to estimate the underlying distribution of the embeddings outperforms other backends such as using cosine similarity to the class means. Unless stated otherwise, all GMMs are realized using scikit-learn [23], initialized with the learned mean values of the sub-cluster AdaCos loss and have a regularized covariance matrix by adding 10^{-3} to the diagonal. To calculate the anomaly scores, two different strategies for the source and target domain are used.

For the source domain, one GMM is trained for each normal data of the source domain belonging to a section and another GMM is trained for each normal data of the source domain belonging to different attribute information. Let $x \in \mathbb{R}^{128}$ denote an embedding, $s(x) \in \mathcal{S}$ denote its section and $a(s(x)) \subset \mathcal{A}$ denote all attribute information that are present in this section. Then, the anomaly score $Z_{\text{source}}(x)$ for x is the given by

$$\begin{aligned} Z_{\text{source}}(x) := &- \max_k \log P(x|s(x),k) \\ &- \max_k \max_{a \in a(s(x))} \log P(x|a,k) \end{aligned} \tag{1}$$

where $P(.|s,k)$, $P(.|a,k)$ denote the weighted likelihoods of component k of the GMMs trained for section $s \in \mathcal{S}$ and target information $a \in \mathcal{A}$, respectively.

For the target domain, the same GMMs trained on the normal data of the target domain belonging to single sections are used. Furthermore, another GMM with three components is trained on the three target samples and thus the cosine distance to the closest normal target sample is also utilized. Using the same notation from before, the anomaly score $Z_{\text{target}}(x)$ for embedding x is given by

$$\begin{aligned} Z_{\text{target}}(x) := &- \max_k \log P(x|s(x),k) \\ &- \max_{k=1,2,3} \log P(x|X_{\text{target}}(s(x)),k) \end{aligned} \tag{2}$$

where $X_{\text{target}}(s(x)) \subset \mathbb{R}^{128}$ denotes the normal training samples of the target domain belonging to the section of x.

In [11], it has been shown that a simple representation derived from the input features leads to surprisingly good performance for the machine type “valve” on the DCASE 2020 dataset. This is the reason why an additional term based on the temporal maximum of the log-Mel spectrogram, denoted by $t_{\max}(x) \in \mathbb{R}^{128}$ for embedding x, is introduced when calculating the anomaly score for the source domain of the machine type “valve”. To this end, a GMM with a single Gaussian component is trained and the altered anomaly score $\widetilde{Z}_{\text{source}}(x)$ for embedding x belonging to machine type “valve” is given by

$$\widetilde{Z}_{\text{source}}(x) := Z_{\text{source}}(x) - \max_{a \in a(s(x))} \log P_{t_{\max}}(t_{\max}(x)|a) \tag{3}$$

where $P_{t_{\max}}(.|a)$ denotes the weighted likelihoods of the single Gaussian trained on the temporal maxima of the log-Mel spectrograms belonging to target information $a \in \mathcal{A}$.

2.4. Ensembling strategy

As done in [11], the proposed neural network for extracting the embeddings is trained with a different number of sub-clusters ranging from 2^0 to 2^4. The same value is used for both sub-cluster AdaCos losses. Thus, there are 5 differently trained versions of each network to extract embeddings. Furthermore, after each 100 epochs of training, the embeddings are extracted and GMMs are trained to calculate the anomaly detection scores. Then, all of these scores are summed-up resulting in 4 subsystems for each network with a specified number of sub-clusters and hence an ensemble consisting of a total of $4 \times 5 = 20$ models.

In addition to that, the described ensembling procedure is repeated by using only a single sub-cluster AdaCos loss classifying among the sections and machine types only and thus removing the second sub-cluster AdaCos loss. This led to slightly better performance for some machine types and to slightly worse performance for other machine types. To obtain anomaly scores for each machine type, the single system is used that led to better performance for the given machine type. More concretely, for the machine types "ToyCar", "ToyTrain", "pump" and "slide rail" the anomaly scores obtained by using the model trained on both losses are used and for "fan", "gearbox" and "valve" the anomaly scores obtained with the models trained on only a single loss are used.

2.5. Setting decision thresholds

Next, it is described how decision thresholds for deciding whether a given test sample is normal or anomalous solely based on the anomaly score are obtained. To this end, the 90th percentile of the anomaly scores of all normal training samples belonging to a given section and a given domain is calculated. Then, all anomaly scores of test samples belonging to the same section and domain that are above this threshold are marked as *anomalous*. For the source domain, $Z_{\text{source}}(x)$ as defined in Eq. (1) is used but for the target domain, only the first term of $Z_{\text{target}}(x)$ is used. The reason is that the likelihoods from the second term belonging to the training data are inappropriately high since the three means of the corresponding GMM are initialized as the three training samples. Hence, when also using the second term of Eq. (2) the decision threshold would also be estimated too high and thus all test data samples belonging to the target domain would be considered anomalous.

3. RESULTS

The results obtained with the proposed system compared to the two baseline systems can be found in Tab. 2. It can be seen that the proposed system significantly outperforms both baseline systems, which both have roughly the same overall performance, on the source and target domains. However, the improvement in terms of AUC is much greater than for pAUC. For nearly all dataset splits the proposed system has a higher AUC than both baseline systems. But for some dataset splits the MobileNetV2-based baseline system has a higher pAUC than the proposed system. For the the machine type "gearbox" the harmonic mean of all pAUCs belonging to the proposed system is even slightly worse than the harmonic mean of the MobileNetV2-based baseline system.

4. SUBMISSIONS

In total, the results obtained with four systems have been submitted to the challenge. More concretely, the results obtained with the proposed system as previously described and three slight variations of it have been submitted. The first variation only consists of the subsystem of the ensemble trained with both sub-cluster AdaCos losses. The second variation is using the mean of the scores of both subsystems instead of using the best-performing subsystem for each of the machine types. And the third variation is the proposed system without also using the simple temporal max-representation for the machine type "valve", i.e. not using the altered anomaly score given in Eq. (3) but the one given in Eq. (1).

5. CONCLUSIONS

In this work, an anomalous sound detection system based on the sub-cluster AdaCos loss function for domain shifted conditions has been presented. The proposed system consists of multiple discriminatively trained neural networks for extracting embeddings from log-Mel spectrograms and utilizes multiple GMMs for estimating distributions of the normal embeddings. These estimated distributions are then used to calculate log-likelihoods for test data and combine them into actual anomaly scores. To decide whether a given test sample is anomalous or normal, individual decision thresholds for each section are computed by taking the 90th percentile from the log-likelihoods of the corresponding normal training samples. In experimental evaluations conducted on the dataset of task 2 of the DCASE challenge 2021, it has been shown that the proposed system significantly outperforms both baseline systems of the challenge in terms of AUC and pAUC on source and target domains of the development set.

6. REFERENCES

[1] S. Kapka, "ID-conditioned auto-encoder for unsupervised anomaly detection," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 71–75.

[2] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Conformer-based sound event detection with semi-supervised learning and data augmentation," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 100–104.

[3] T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. Hoang Trong, D. Wood, M. Tatsubori, and R. Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 66–70.

[4] 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.

[5] Q. Zhou, "ArcFace based sound mobilenets for DCASE 2020 task 2," DCASE2020 Challenge, Tech. Rep., 2020.

[6] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 96–99.

[7] K. Wilkinghoff, "Using look, listen, and learn embeddings for detecting anomalous sounds in machine condition monitoring," in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, 2020, pp. 215–219.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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 sound detection for machine condition monitoring under domain shifted conditions,” *In arXiv e-prints: 2106.04492, 1–5*, 2021.

[13] 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,” *arXiv preprint arXiv:2106.02369*, 2021.

[14] R. Tanabe, H. Purohit, K. Dohi, T. Endo, Y. Nikaido, T. Nakamura, and Y. Kawaguchi, “MIMII DUE: Sound dataset for malfunctioning industrial machine investigation and inspection with domain shifts due to changes in operational and environmental conditions,” *In arXiv e-prints: 2006.05822, 1–4*, 2021.

[15] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, “ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection,” in *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 313–317.

[16] H. Purohit, R. Tanabe, T. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, “MIMII Dataset: Sound dataset for malfunctioning industrial machine investigation and inspection,” in *Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. New York University, 2019, pp. 209–213.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

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

[22] 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.

[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 2: AUCs and pAUCs per machine type obtained with the baseline system and the proposed system. 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	67.63%	51.87%	66.56%	66.47%	79.15%	61.63%
ToyCar	1	source	61.97%	51.82%	71.58%	66.44%	91.44%	71.26%
ToyCar	2	source	74.36%	55.56%	40.37%	47.48%	96.89%	85.05%
ToyCar	0	target	54.50%	50.52%	61.32%	52.61%	92.14%	80.89%
ToyCar	1	target	64.12%	51.14%	72.48%	63.99%	77.10%	63.26%
ToyCar	2	target	56.57%	52.61%	45.17%	48.85%	62.63%	57.95%
ToyCar	harmonic mean		62.49%	52.36%	56.04%	56.37%	81.43%	68.62%
ToyTrain	0	source	72.67%	69.38%	69.84%	54.43%	96.51%	91.11%
ToyTrain	1	source	72.65%	62.52%	64.79%	54.09%	89.02%	77.79%
ToyTrain	2	source	69.91%	47.48%	69.28%	47.66%	87.91%	47.37%
ToyTrain	0	target	56.07%	50.62%	46.28%	51.27%	72.29%	48.37%
ToyTrain	1	target	51.13%	48.60%	53.38%	49.60%	50.86%	49.89%
ToyTrain	2	target	55.57%	50.79%	51.42%	53.40%	94.67%	79.84%
ToyTrain	harmonic mean		61.71%	53.81%	57.46%	51.61%	77.89%	61.11%
fan	0	source	66.69%	57.08%	43.62%	50.45%	73.41%	62.26%
fan	1	source	67.43%	50.72%	78.33%	78.37%	89.02%	84.53%
fan	2	source	64.21%	53.12%	74.21%	76.80%	84.01%	76.32%
fan	0	target	69.70%	55.13%	53.34%	56.01%	55.37%	48.47%
fan	1	target	49.99%	48.49%	78.12%	66.41%	87.94%	75.53%
fan	2	target	66.19%	56.93%	60.35%	60.97%	71.31%	70.68%
fan	harmonic mean		63.24%	53.38%	61.56%	63.02%	74.80%	67.41%
gearbox	0	source	56.03%	51.59%	81.35%	70.46%	85.25%	73.93%
gearbox	1	source	72.77%	52.30%	60.74%	53.88%	85.91%	54.05%
gearbox	2	source	58.96%	51.82%	71.58%	62.23%	59.31%	48.41%
gearbox	0	target	74.29%	55.67%	75.02%	64.77%	87.62%	71.61%
gearbox	1	target	72.12%	51.78%	56.27%	53.30%	86.87%	56.85%
gearbox	2	target	66.41%	53.66%	64.45%	55.58%	65.41%	52.96%
gearbox	harmonic mean		65.97%	52.76%	66.70%	59.16%	76.49%	58.19%
pump	0	source	67.48%	61.83%	64.09%	62.40%	77.15%	63.53%
pump	1	source	82.38%	58.29%	86.27%	66.66%	98.14%	90.47%
pump	2	source	63.93%	55.44%	53.70%	50.98%	79.15%	65.68%
pump	0	target	58.01%	51.53%	59.09%	53.96%	58.54%	51.21%
pump	1	target	47.35%	49.65%	71.86%	62.69%	87.89%	61.37%
pump	2	target	62.78%	51.67%	50.16%	51.69%	73.57%	57.74%
pump	harmonic mean		61.92%	54.41%	61.89%	57.37%	77.08%	63.05%
slide rail	0	source	74.09%	52.45%	61.51%	53.97%	95.56%	82.11%
slide rail	1	source	82.16%	60.29%	79.97%	55.62%	94.28%	71.58%
slide rail	2	source	78.34%	65.16%	79.86%	71.88%	84.05%	76.59%
slide rail	0	target	67.22%	57.32%	51.96%	51.96%	81.57%	59.47%
slide rail	1	target	66.94%	53.08%	46.83%	52.02%	65.97%	49.84%
slide rail	2	target	46.20%	50.10%	55.61%	55.71%	73.40%	58.00%
slide rail	harmonic mean		66.74%	55.94%	59.26%	56.00%	81.07%	64.29%
valve	0	source	50.34%	50.82%	58.34%	54.97%	79.54%	62.54%
valve	1	source	53.52%	49.33%	53.57%	50.09%	91.02%	67.53%
valve	2	source	59.91%	51.96%	56.13%	51.69%	98.09%	92.00%
valve	0	target	47.12%	48.68%	52.19%	51.54%	68.91%	63.79%
valve	1	target	56.39%	53.88%	68.59%	57.83%	80.02%	63.26%
valve	2	target	55.16%	48.97%	53.58%	50.86%	78.30%	57.05%
valve	harmonic mean		53.41%	50.54%	56.51%	52.64%	81.60%	66.16%
all	harmonic mean		61.93%	53.27%	59.72%	56.37%	78.54%	63.93%