SELF-SUPERVISED ACOUSTIC ANOMALY DETECTION VIA CONTRASTIVE LEARNING

Hadi Hojjati and Narges Armanfard

Department of Electrical and Computer Engineering, McGill University
Mila - Quebec AI Institute, Montreal, QC, Canada

ABSTRACT

We propose an acoustic anomaly detection algorithm based on the framework of contrastive learning. Contrastive learning is a recently proposed self-supervised approach that has shown promising results in image classification and speech recognition. However, its application in anomaly detection is underexplored. Earlier studies have demonstrated that it can achieve state-of-the-art performance in image anomaly detection, but its capability in anomalous sound detection is yet to be investigated. For the first time, we propose a contrastive learning-based framework that is suitable for acoustic anomaly detection. Since most existing contrastive learning approaches are targeted toward images, the effect of other data transformations on the performance of the algorithm is unknown. Our framework learns a representation from unlabeled data by applying audio-specific data augmentations. We show that in the resulting latent space, normal and abnormal points are distinguishable. Experiments conducted on the MIMII dataset confirm that our approach can outperform competing methods in detecting anomalies. [1] [2]

***Index Terms*—** Contrastive Learning, Anamolous Sound Detection, Anomaly Detection, Self-Supervised Learning

1. INTRODUCTION

In view of the machine learning's rapid development, anomaly detection algorithms have emerged as efficient tools for monitoring the operation of industrial equipment and early detection of machine failure. Specifically, methods that can detect anomalies in the visual domain have dominated the industrial anomaly detection field [1]. In some applications, however, visual inspections cannot detect a machine's defects. A possible solution is to incorporate acoustic monitoring and anomalous sound detection algorithms for identifying the issues that cannot be revealed by cameras [2].

Comparing to most common machine learning tasks, anomaly detection faces unique challenges, such as lack of labeled anomaly samples and imbalanced training datasets. A flurry of models, including one-class classifiers, autoencoders, generative adversarial networks, and self-supervised techniques have been proposed for tackling these issues [3, 4]. The underlying assumption behind most of these methods is that we only have access to the normal samples during the training phase. Moreover, they assume that a network which is trained on the normal data will perform poorly on abnormal samples and thus can be used as an anomaly detector [5].

Contrastive learning has attracted an immense amount of attention from the machine learning community over the past few years. The basic idea behind it is to pull representations of different views of the same sample closer together, while pushing them away from other samples of the batch [6]. The success of contrastive learning has been marked by the introduction of SimCLR algorithm for self-supervised image classification. Chen et al. [6] have shown that self-supervised SimCLR could match the performance of a supervised ResNet-50 without using any training labels.

Several studies have evaluated the efficiency of contrastive learning for anomaly detection and have demonstrated that it can significantly improve the accuracy on benchmark datasets. Tack et al. [7] came up with the idea of using shifting transformations for generating negative pairs, and showed that this approach would outperform other algorithms. In another recent study, Sehwag et al. [8] proposed an an unsupervised anomaly detection approach based on the framework of contrastive learning. They also extended their work to few-shot anomaly detection and manifested that by using only a few anomaly samples during training, their algorithm can achieve a state-of-the-art performance.

The efficiency of contrastive learning algorithms heavily depends on the domain-specific augmentations. Existing contrastive learning-based models for anomaly detection are commonly focused on images and videos. Although these methods have significantly improved the results in those domains, they cannot be applied to other data types, such as audio, since they use geometric transformations like rotation and cropping.

In this paper, we propose an acoustic anomaly detection framework by employing audio-specific augmentations and contrastive loss. Similar to other state-of-the-art algorithms, our approach assumes that we only have access to normal

[2] Code is available at: https://github.com/Armanfard-Lab/AADCL

training data. To the best of our knowledge, this is the first study that employs contrastive learning for acoustic anomaly detection. Inline with earlier studies on images, we show that contrastive learning is a powerful tool for acoustic anomaly detection, and can achieve state-of-the-art performance.

2. METHOD

An overview of our anomaly detection framework is depicted in Fig. 1.

Contrastive learning can learn patterns from unlabeled data by creating different instances of a sample with the aid of data augmenting transformations. These transformations should preserve the underlying information of data while not being redundant to the original sample so that the network can learn to detect its distinguishing features.

Let $\mathcal{T}$ denotes the family of possible augmentation operators. Similar to SimCLR [6] we stochastically apply two operators $t_1 \sim \mathcal{T}$ and $t_2 \sim \mathcal{T}$ to generate two correlated views from each sample in the batch: $\tilde{x}_1^{(k)} = t_1(x^{(k)})$ and $\tilde{x}_2^{(k)} = t_2(x^{(k)})$.

We utilized the following transformation to generate augmented versions of a given sample:

- **Pitch Shift:** This transformation randomly increases or decreases the pitch of the audio. The pitch shift is chosen from the range of $[-10, 10]$ semitones with even probability. This range, and the ranges of all other augmentations' parameters, are empirically found by experimenting with different values and finding the ones that does not distort the signal completely.
- **Time Stretch:** It changes the speed of the audio by a predefined rate. If the rate is smaller than one, the audio will be slowed down and otherwise, it will speed up. We re-sample the resulting audio to get a signal with the same length as of the input. The rate of the time-stretch is randomly chosen from the range of $[0.5, 2]$.
- **White Noise Injection:** This module injects a white gaussian noise to the data. The intensity of the noise is determined by the signal-to-noise (SNR) ratio that we specify. In this work, we randomly choose the SNR from the range of $[-6, 6]$.
- **Fade In/Fade Out:** Adds a fade in or a fade out to the beginning and the end of the signal. The fade type can be linear, logarithmic, exponential, quarter sinusoidal, or half sinusoidal. We randomly apply each fade type with uniform probability. The size of the fade is also randomly chosen equal to a value between zero and half of the signal's length.
- **Time Shifting:** Shifts the audio signal forward or backward. The degree of the shift is randomly chosen from the range of zero and half of the signal's length.
- **Time Masking:** This transformation randomly selects a segment of the signal and set it equal to zero or another constant value. The size of the masked portion of the audio is randomly chosen to a value less than $\frac{1}{10}$ of the signal's length.
- **Frequency Masking:** It applies a random masking to the frequency spectrum of the signal. In other words, it randomly removes a segment of frequencies of the audio. The length of the masked segment is randomly chosen to a value less than $\frac{1}{10}$ of the signal's frequency length.

After applying the transformations, we extract the Mel spectograms of raw audio signals. Mel spectograms is a time-frequency representation of signals which is inspired by human hearing perception, and is a standard feature for audio analysis. We feed the resulting two-dimensional mel spectograms to a base encoder $f(.)$, which is a neural network that maps the input data to a lower-dimensional vector $h^{(k)} = f(\tilde{x}^{(k)})$. The choice of the encoder is arbitrary but for simplicity, we used ResNet-18 in all our analyses. Although ResNet-18 has been originally adopted for natural image classification, earlier studies have shown its efficiency for analyzing spectograms as they possess features that are similar to natural images [9].

Following the original SimCLR paper [6], we apply an additional projection head $g(.)$ that maps the latent representation, h, to a subspace where the contrastive loss is calculated. $g(.)$ is a Multi Layer Perceptron (MLP) with one hidden layer and ReLU activation function, and is discarded during the inference phase.

The contrastive loss is applied to the output of the projection head, $z^{(k)} = g(h^{(k)})$. It aims to pull together the representation of positive pairs $(z_i^{(k)}, z_j^{(k)})$, while pushing them away from negative pairs. We simply consider all other samples in the batch as negative pairs.

We used Normalized Temperature-scaled Cross-Entropy loss (NT-Xent) as our contrastive loss function [10]. The NT-Xent loss for the positive pair (i, j) is calculated as follows:

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\mathrm{z_i}, \mathrm{z_j})/\tau)}{\sum_{a=1}^{2N} \mathbb{1}_{[\mathrm{a}\neq\mathrm{i}]} \exp(\mathrm{sim}(\mathrm{z_i}, \mathrm{z_a})/\tau)} \tag{1}$$

In the above formula, $\mathrm{sim}(\mathrm{z_i}, \mathrm{z_j})$ denotes cosine similarity between z_i and z_j, and is equal to $\mathrm{sim}(\mathrm{z_i}, \mathrm{z_j}) = \frac{z_i . z_j}{\|z_i\| \|z_j\|}$. The total loss of each batch can be calculated as:

$$\mathcal{L}_{NTXent} = \frac{1}{2N} \sum_{i=1}^{N} \ell_{i,j} + \ell_{j,i} \tag{2}$$

In self-supervised learning, it is common to train the network for an auxiliary task in order to improve its performance on the main task [11]. One common proxy task is to train a

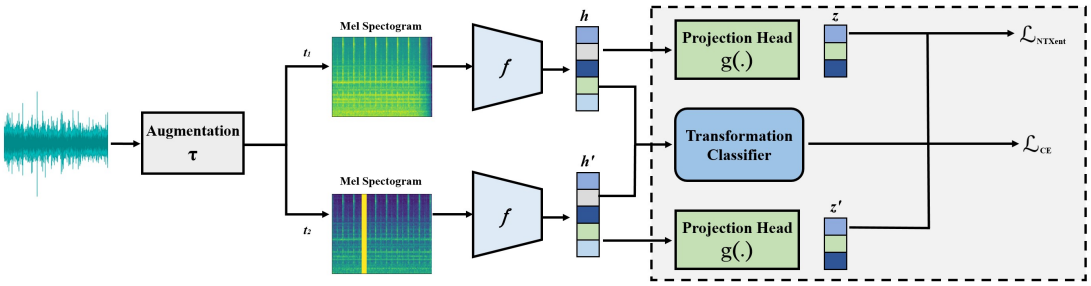


Fig. 1. An overview of the training phase of our proposed method.

simple classifier on top of the latent representation to predict the type of the transformation that was applied to data in the first place. Therefore, we also added a simple linear classifier, $p_{cls}(\mathcal{T} = t_i|\tilde{x})$ for predicting the applied transformation on top of the encoder's output. Multi-class cross entropy is chosen as the loss function of this classifier, $\mathcal{L}_{CE-cls}$. The final loss of our network will be the weighted sum of the contrastive loss and the loss of the transformation classifier:

$$\mathcal{L}_{total} = \mathcal{L}_{NTXent} + \lambda \mathcal{L}_{CE-cls}. \quad (3)$$

Where λ is a balancing hyperparameter. We set $\lambda = 0.1$ throughout all our experiments.

Finally, we need a scoring function that maps the latent representation to a scalar that quantifies the degree of sample's abnormality. To this end, we utilize the Mahalanobis distance [12]. Mahalonobis distance is a metric that measures the distance between a point and a distribution.

During the inference, we measure the distance between the latent embedding of the query sample and the representation of normal training instances. For a given input x, the anomaly score S_x is calculated as follows:

$$S_x = (h_x - \mu)^T \Sigma^{-1} (h_x - \mu) \quad (4)$$

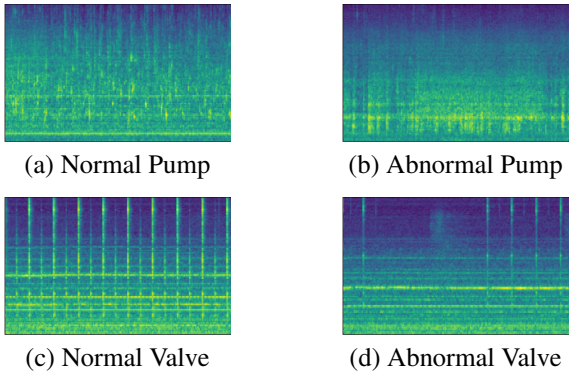
Where μ is the mean vector of normal training samples, Σ is their covariance matrix, and $h_x = f(x)$.

3. EXPERIMENTS

To gain an insight into the behavior and performance of our proposed framework, we carried out several experiments using an industrial anomalous sound detection dataset.

3.1. Dataset and Task

MIMII [13] is a real-life dataset that contains audio samples for detecting malfunctioning industrial machinery, and has been used by similar papers for performance assessment [2][14]. It contains 10-seconds audio segments from four machine types: Fans, Pumps, Slide-Rails, and Valves. The signals are recorded with a 16 KHz sampling frequency.



(a) Normal Pump (b) Abnormal Pump

(c) Normal Valve (d) Abnormal Valve

Fig. 2. Examples of normal and abnormal samples in the dataset.

Readers can refer to [13] for a more detailed explanation of the dataset and experimental condition.

In the publicly released dataset, each machine type contains four IDs. For each machine ID, the normal and abnormal samples are available. Some examples of normal and abnormal samples are illustrated in Fig. 2. We first split the normal data to train and test sets. Then we concatenate the test normal data with all abnormal samples to create the final test dataset. For the training, we only use the normal train dataset. We train and test the network on each individual machine IDs.

3.2. Implementation

For extracting the Mel-spectogram, we set the number of Mel filters equal to $M = 128$. We also set the hop length and number of FFT points to 512 and 2048 respectively.

As earlier mentioned, we used ResNet-18 as our base encoder [15]. We modified the size of the final layer and set it to 512. For the projection head, we use an MLP with a hidden layer of size 256 and representation vector of length 128.

We used the ADAM optimizer [16] with initial learning rate equal to $\eta = 0.01$ as our optimizer. The batch size is set to $N = 128$ which would be equal to $\tilde{N} = 256$ after applying the transformations. We trained the network for $n = 400$ epochs. Finally, we fixed the temperature constant to $\tau =$

Algorithm	Fan	Pump	Slide Rail	Valve
AE Baseline	63.24%	61.92%	66.74%	53.41%
IDNN	64.4%	61.48%	67.80%	57.37%
VIDNN	66.5%	60.08%	67.6%	59.0%
FREAK	62.2%	62.4%	66.4%	56.5%
Our Method	**80.11%**	**70.12%**	**77.43%**	**84.17%**

Table 1. Average AUC (%) of our method, in comparison with the results of a baseline AE, IDNN, VIDNN, and FREAK [14].

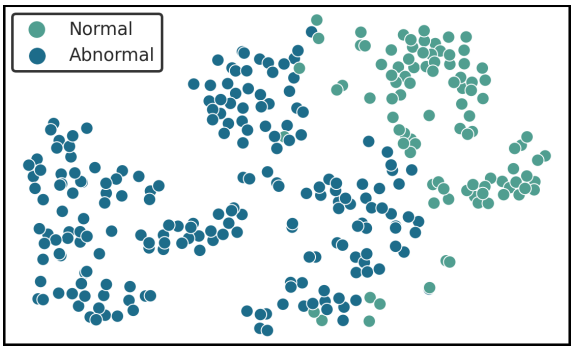


Fig. 3. Latent embedding T-SNE projection for the test dataset of valve ID 0.

0.07.

We evaluated the performance based on the Area Under the Curve (AUC) of the receiver operating characteristics (ROC). We calculated the AUC five times for each machine ID and averaged the results.

The code was implemented in Python using PyTorch, Torchaudio and Librosa [17] libraries.

3.3. Results

The average AUC over all individual IDs for different machine types is reported in Table 1. To better evaluate the performance of our model, we also included the results of several other competing algorithms [14].

As is shown in Table 1, our proposed approach has achieved significantly improved performance in identifying anomalies, and beats other competing algorithms. In accordance with previous works [7], these results confirm that in general, contrastive learning is a powerful tool for anomaly detection. Earlier works have shown that contrastive learning can significantly improve the state-of-the-art performance in image anomaly detection and the result of our work can further extend its success to acoustic anomaly detection.

Intuitively, we can describe the effectiveness of the learned representation in detecting anomalies by considering the loss function of our network. During training, we force our network to map the augmented normal training samples far from each other in the latent space. Since all our training samples are normal, they share similar features but instead, we enforce the network to focus on the minor discrepancies that exist between them. Many anomalies also share lots of similar features with normal samples and we can only identify them by paying attention to small inconsistencies.

Therefore, when the network is fed with a new sample, it identifies the features that differ from the latent distribution of the normal class. We assume that if the sample belongs to the normal class, there is a higher probability that a point with similar features has been seen before during training. In other words, the new data will be mapped closer to the latent embedding of normal points if it belongs to the normal class. In contrast, if the point is abnormal, it is more likely that the network has not seen a similar point during training and will map it further from the distribution of the normal data.

To better understand the behavior of the latent representation, we employed T-SNE projection [18] to visualize the latent embedding of one of the machines in Fig. 3. Looking at this figure, we can confirm that anomalies and normal points are mapped to separate clusters and are distinguishable in the latent space.

Another remarkable observation is that our model achieves its best performance on the data of valves, while other algorithms almost fail to identify anomalies in this machine. An example of valve's normal and abnormal data is illustrated in Fig. 2 (c) and (d). We can see that there is a repetitive pattern in the normal spectogram of valves and a deviation from this pattern can be an indicator of anomaly. Based on the results of Table 1, the competing algorithms which are based on autoencoders and reconstruction error, cannot efficiently capture these anomaly types. This means that our approach can be useful in some cases where other algorithms, such as autoenders, fail to identify anomalies.

It is noteworthy to mention that we can possibly improve the results of the network if we assume that we have access to a limited number of labeled anomalies. We can use these samples for tuning an anomaly detector on top of the latent representation, or even incorporate them for learning a better representation by adding relevant terms to the loss function.

4. CONCLUSION

In this paper, we proposed an acoustic anomaly detection algorithm that employs contrastive learning. Experimental results show that our proposed approach can significantly outperform competing models. Combined with the outcome of the previous works on image anomaly detection, these results show the effectiveness of contrastive learning for anomaly detection. As a future expansion to the current work, one can investigate the performance under the assumption that a few labeled anomalies are present in the training dataset.

5. REFERENCES

[1] Benjamin Staar, Michael Lütjen, and Michael Freitag, "Anomaly detection with convolutional neural networks for industrial surface inspection," *Procedia CIRP*, vol. 79, pp. 484–489, 2019, 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy.

[2] Kaori Suefusa, Tomoya Nishida, Harsh Purohit, Ryo Tanabe, Takashi Endo, and Yohei Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 271–275.

[3] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Müller, "A unifying review of deep and shallow anomaly detection," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 756–795, 2021.

[4] Hadi Hojjati and Narges Armanfard, "Dasvdd: Deep autoencoding support vector data descriptor for anomaly detection," 2021.

[5] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, Hal Daumé III and Aarti Singh, Eds. 13–18 Jul 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 1597–1607, PMLR.

[7] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin, "Csi: Novelty detection via contrastive learning on distributionally shifted instances," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 11839–11852, Curran Associates, Inc.

[8] Vikash Sehwag, Mung Chiang, and Prateek Mittal, "Ssd: A unified framework for self-supervised outlier detection," in *International Conference on Learning Representations*, 2021.

[9] Quan Zhou, Jianhua Shan, Wenlong Ding, Chengyin Wang, Shi Yuan, Fuchun Sun, Haiyuan Li, and Bin Fang, "Cough recognition based on mel-spectrogram and convolutional neural network," *Frontiers in Robotics and AI*, vol. 8, pp. 112, 2021.

[10] Kihyuk Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. 2016, vol. 29, Curran Associates, Inc.

[11] Kota Dohi, Takashi Endo, Harsh Purohit, Ryo Tanabe, and Yohei Kawaguchi, "Flow-based self-supervised density estimation for anomalous sound detection," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 336–340.

[12] P.C. Mahalanobis, "On the generalised distance in statistics," in *Proceedings of the National Institute of Sciences of India*, 1936, vol. 2, pp. 49–55.

[13] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi, "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, New York University, NY, USA, October 2019, pp. 209–213.

[14] Haishen Lu, Yujie Fu, Huajing Qin, Shijin Huang, Yihan Wang, Chen Deng, Tianchu Yao, Huitian Jiang, Haifeng Wen, and Chuang Shi, "Anomalous sounds detection using autoencoder and classification methods," Tech. Rep., DCASE2021 Challenge, July 2021.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[16] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, 2017.

[17] Brian McFee et al., "librosa/librosa: 0.8.1rc2," May 2021.

[18] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.