UNSUPERVISED DETECTION OF ANOMALOUS SOUND FOR MACHINE MONITORING UNDER DOMAIN SHIFTED CONDITION BASED ON GANS AND AUTOENCODERS

Technical Report

Amirhossein Hassankhani, *Afshin Dini*, *Konstantinos Drossos*

Audio Research Group, Tampere University, Finland
{amirhossein.hassankhani, afshin.dini, konstantinos.drossos}@tuni.fi

ABSTRACT

This report presents an unsupervised method for detecting anomalous industrial machine sounds, taken under two different conditions and shifted domains, and submitted to DCASE 2021 Task 2. The method tries to map the distribution of data into a learned latent space, using a reconstructive autoencoder followed by an additional second encoder. Furthermore, the method employs a discriminator trying to differentiate between the input and the reconstructed audio to and from the autoencoder. All components are jointly optimized, using a sum of weighted losses and utilizing an adversarial setting between the autoencoder and the discriminator. Anomaly is detected through the distance between the output of the two encoders. Obtained results show that the method performs better than the provided baseline in some cases.

Index Terms— Anomaly detection, generative adversarial network, domain adaptation, GAN, autoencoder

1. INTRODUCTION

Nowadays, detection of anomalous sound for detecting machine conditions becomes one of the most important issues in monitoring and diagnosing fault in industrial systems. Deep learning methods, and specifically unsupervised learning based ones, are widely used to deal with this problem as the types of anomalies are usually undefined and unknown in real world applications.

In general, unsupervised audio anomaly detection methods are divided into three categories. The first category seems to include the vast majority of methods, which are reconstruction-based ones such as dense and convolutional autoencoders [1, 2, 3], and generative adversarial network (GAN) based anomaly detection methods [4]. These methods find the anomalies by thresholding reconstruction error between the input and the output of the employed autoencoder, by firstly mapping the input audio signals into a learned and low dimensional space, and then try to reconstruct the input audio signals from the latent, low-dimensional space data. It is assumed that the anomaly cannot be reconstructed appropriately by the latent representation of the input data, as a result of which the reconstruction error can be used for anomaly detection [5]. Though, the efficiency of the methods in this first category is restricted since finding the appropriate low-dimensional data space is difficult in some applications [6]. Moreover, these approaches cannot sometimes deal with the unbalanced data gathered from devices in different environments [7].

The second category of methods consists of one class classification approaches such as deep support vector data description (SVDD) [8, 9] which tries to find a boundary around non-anomalous instances and label the anomaly data outside this area. The third category contains clustering methods such as k-means clustering and Gaussian mixture models which cluster the data and detect the anomalies according to their distance [10, 11, 12]. It is reported that these distance-based anomaly detection methods cannot be used with high dimensional data due to curse of dimensionality [13].

According to previous work [14], architectures which use GANs alongside of autoencoders are able to address some of the above issues which in turn will end up in better results while dealing with anomaly detection problem. Based on that, we follow the GAN-based method presented in [14], adopt it by having all employed learned processes to be non-symmetric, and apply it on the task of anomalous sound detection under domain shifted condition.

The rest of the report is organized as follows. In Section 2 is described the proposed method and Section 3 describes the experimental setup. The obtained results are presented in Section 4 and Section 5 concludes the report.

2. PROPOSED METHOD

Our method consists of two encoders and one decoder, takes as an input an audio signal, and outputs an indication of whether this signal contains anomalous sound, regarding a specific device. To optimize the encoders and the decoder, we additionally employ a discriminator, a GAN setting, and three losses, and we utilize a dataset of only non-anomalous sounds. During training, the first encoder takes as an input the non-anomalous sound and the decoder tries to reconstruct the input audio from the output of the first encoder. The second encoder, tries to reconstruct the output of the first encoder, using the output of the decoder. The discriminator tries to differentiate between original input audio and the output of the decoder. The training happens jointly, minimizing corresponding losses for the above processes. After optimization, we use the output of the encoders to determine if the input is anomalous audio or not. An illustration of our method is in Figure 1 and the code for the implementation is available online[1].

Specifically, we follow the method presented in [14] and we employ an encoder AE_{E1}, a decoder AE_D, an additional encoder AE_{E2}, and a discriminator D. The input to AE_{E1} is a sequence of T vectors with F audio features, $\mathbf{X} \in \mathbb{R}^{T \times F}$. AE_{E1} consists of sequential 2D convolutional blocks, where each block consists

[*] Equally contributing authors.

[1] https://github.com/afshindini/DCASE_2021_Challenge_Task2.git

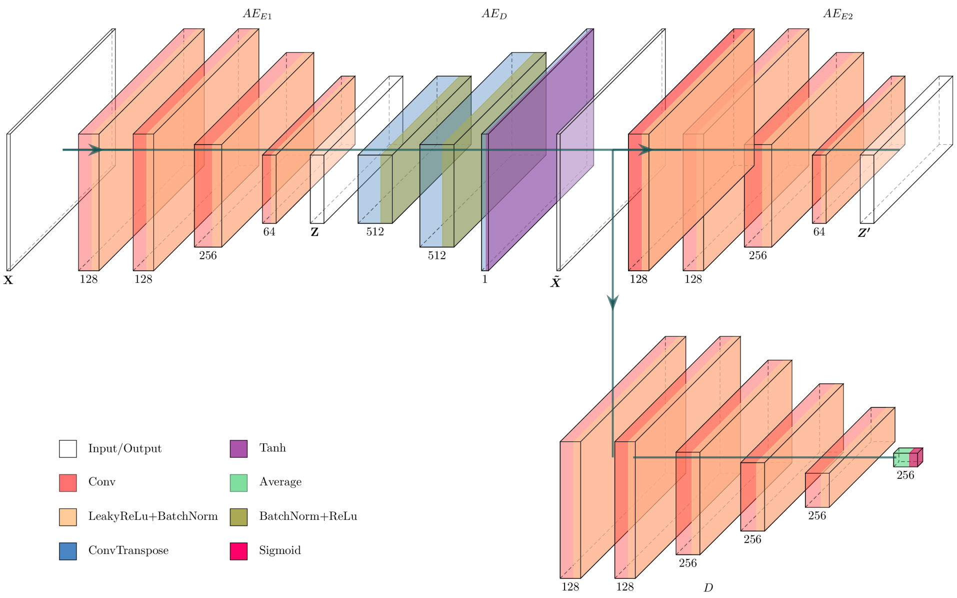

Figure 1: Our proposed method

of a 2D convolutional neural network (CNN), followed by a leaky rectified linear unit (LeakyReLU) non-linearity, and a batch normalization. AE_{E1} processes sequentially the input $\mathbf{X}$ and outputs an encoded representation of $\mathbf{X}$

$$\mathbf{Z} = AE_{E1}(\mathbf{X}), \tag{1}$$

where $\mathbf{Z} \in \mathbb{R}^{T' \times F'}$, with $T' < T$ and $F' < F$. The target of AE_{E1} is to model the underlying distribution of the input, non-anomalous data, according to the typical set-up of an autoencoder under a reconstruction objective [15].

Then, AE_D takes as an input the $\mathbf{Z}$ and outputs a reconstructed version of $\mathbf{X}$, $\tilde{\mathbf{X}}$. AE_D consists of a series of transposed 2D CNNs, each one followed by a rectified linear unit (ReLU) and a batch normalization, except the last CNN which is followed only by a tanh non-linearity. AE_D processes $\mathbf{Z}$ as

$$\tilde{\mathbf{X}} = AE_D(\mathbf{Z}), \tag{2}$$

where $\tilde{\mathbf{X}} \in \mathbb{R}^{T \times F}$. The target of AE_D is to learn to reconstruct back the original input $\mathbf{X}$, based on the information learned from AE_{E1}. $\tilde{\mathbf{X}}$ is given as an input to the second encoder, AE_{E2}, which has the exact same hyper-parameters and architecture as AE_{E1}. The purpose of using AE_{E2} is to have a second encoded form of $\mathbf{X}$, in order to compare with the output of AE_{E1}. The hypothesis behind that, and according to [14], is that the reconstructing autoencoder AE_{E1} and AE_D, will learn to model the underlying distribution of the non-anomalous data. Thus, any shift from that distribution, will be outlined by the usage of AE_{E2}. The output of AE_{E2} is calculated as

$$\mathbf{Z}' = AE_{E2}(\tilde{\mathbf{X}}). \tag{3}$$

To enforce that $\tilde{\mathbf{X}}$ will conform to the original distribution of $\mathbf{X}$, we employ D which is a series of 2D CNNs, each one followed by LeakyReLU and a batch normalization process. The discriminator D tries to differentiate between $\tilde{\mathbf{X}}$ and $\mathbf{X}$, by employing the loss

$$\mathcal{L}_{\text{adv}} = ||\sigma(D(\mathbf{X})) - \sigma(D(\tilde{\mathbf{X}}))||_2. \tag{4}$$

where σ is the sigmoid function and the output dimensionality of D is 1. AE_{E1}, AE_D, and AE_{E2} are optimized by employing the losses

$$\mathcal{L}_{\text{ano}} = ||\mathbf{Z} - \mathbf{Z}'||_2 \text{ and} \tag{5}$$

$$\mathcal{L}_{\text{rec}} = ||\mathbf{X} - \tilde{\mathbf{X}}||_1. \tag{6}$$

We jointly optimize AE_{E1}, AE_D, AE_{E2}, and D, by minimizing the $\mathcal{L}_{\text{tot}}$

$$\mathcal{L}_{\text{tot}} = w_{\text{rec}}\mathcal{L}_{\text{rec}} + w_{\text{ano}}\mathcal{L}_{\text{ano}} + w_{\text{adv}}\mathcal{L}_{\text{adv}}. \tag{7}$$

3. EXPERIMENTAL SETUP

In order to test our method, we apply it on the development dataset of DCASE 2021 challenge [16, 17] and compare the results for source and target domains with the two baseline methods, mentioned in this challenge. The dataset is divided into three splits, namely development, additional, and evaluation. Each split contains seven types of machines as Fan, Slider, Gearbox, Pump, Valve, ToyCar and ToyTrain and each type of machine consists of three sections as machine IDs. Each section also consists of anomalous and non-anomalous data from the two domains. The source domain is the main domain where most of the data are collected from. The target domain is the shifted domain one only a few audio files are provided from this one. All audio clips in the dataset are 10 seconds long and as our input features, $\mathbf{X}$, we use $F = 128$ mel band energies, by utilizing 1024 samples windows with 50% overlap, resulting in $T = 312$.

We train our method separately for each machine type and per each section or machine ID of the development split. Then we evaluate using the 200 hundred non-anomalous and anomalous audio

Table 1: Average AUC for source domain data

	Sections								
Devices	Section 0			Section 1			Section 2		
	AE	MobileNet	Ours	AE	MobileNet	Ours	AE	MobileNet	Ours
Fan	**66.7%**	43.6%	57.5%	67.4%	**78.3%**	54.7%	64.2%	**74.2%**	72.9%
Gearbox	56.0%	**81.4%**	68.8%	**72.8%**	60.7%	70.4%	59.0%	71.6%	**72.8%**
Pump	67.5%	64.9%	**67.6%**	82.4%	**86.3%**	61.5%	**63.9%**	53.7%	62.1%
SlideRail	74.1%	61.5%	**74.4%**	**82.2%**	80.0%	71.1%	78.3%	**79.9%**	78.3%
ToyCar	**67.6%**	66.6%	66.9%	62.0%	**71.6%**	62.6%	**74.4%**	40.4%	64.7%
ToyTrain	**72.7%**	69.8%	44.6%	**72.7%**	64.8%	66.5%	**69.9%**	69.3%	56.5%
Valve	50.3%	**58.3%**	53.9%	53.5%	53.6%	**55.7%**	59.9%	56.1%	**62.1%**

Table 2: Average AUC for target domain data

	Sections								
Devices	Section 0			Section 1			Section 2		
	AE	MobileNet	Ours	AE	MobileNet	Ours	AE	MobileNet	Ours
Fan	**69.7%**	53.3%	68.0%	50.0%	**78.1%**	57.8%	66.2%	60.3%	**69.2%**
Gearbox	74.3%	**75.0%**	71.0%	**72.1%**	56.3%	68.3%	66.4%	64.5%	**68.5%**
Pump	58.1%	59.1%	**60.6%**	47.4%	**71.9%**	55.5%	**62.8%**	50.2%	57.6%
SlideRail	67.2%	52.0%	**68.8%**	**66.9%**	46.8%	62.4%	46.2%	55.6%	**63.5%**
ToyCar	54.5%	**61.3%**	57.9%	64.2%	**72.5%**	66.3%	56.6%	45.2%	**60.3%**
ToyTrain	**56.1%**	46.3%	47.5%	51.3%	53.4%	**56.0%**	55.6%	51.4%	**58.6%**
Valve	47.1%	52.2%	**56.6%**	56.4%	**68.6%**	59.4%	**55.2%**	53.6%	54.9%

clips taken from source and also another 200 hundred audio clips from the target domain, provided by the employed dataset. AUC metric is used for analyzing and comparing the results, as suggested by DCASE 2021 Task 2 [18].

Different architectures with various complexity are considered for each sub network. For AE_{E1} and AE_{E2}, four convolutional layers with kernel size of five, stride of two, and padding of two, are used. For AE_D, a smaller architecture is designed in such a way that only three transposed convolutional networks with kernel size of five, stride of two, and padding of two. D structure is similar to the encoders, however, a more complex network with five convolutional layers, kernel size of five, stride of two, and padding of two, is used. Regarding the total loss, $\mathcal{L}_{\text{tot}}$, we consider same weights for $\mathcal{L}_{\text{avd}}$ and $\mathcal{L}_{\text{ano}}$ ($w_{\text{avd}} = w_{\text{ano}} = 1$) and a larger weight for $\mathcal{L}_{\text{rec}}$ ($w_{\text{rec}} = 50$) since it is important for this architecture to be able to reconstruct $\mathbf{X}$ in every situation and choosing larger weight for w_{rec} will ascertain this objective in much shorter time.

We employ around 300 epochs for minimizing $\mathcal{L}_{\text{tot}}$, using the stochastic gradient descent algorithm (SGD) and a batch size of 64. We use SGD rather than Adam [19] since it is proven theoretically that in most cases SGD generalizes better than Adam, and it is also able to escape better from the flattened local minima than Adam [20].

4. RESULTS

Since the labels for anomalous and non-anomalous data are only specified for the development split, we represent the results of our method over the development split in this report. We compare our results with two baseline methods, mentioned in DCASE 2021 challenge as Autoencoder baseline method and also Mobile V2 approach. From Table 1 and 2, it is obvious that our approach achieves better results in some cases even better than both of the baseline methods. The best result in terms of AUC is highlighted per machine type and machine ID in these tables.

From our experimental process, we see that the asymmetric architecture of $AE.$ and D increases the performance of the method, compared to the original proposal of the method in [14]. Considering a more complex discriminator than encoder and decoder allows it to recognize the reconstructed features from the original ones more precisely and this matter will force the encoder to be trained in such a way that it will be able to find the distribution of the data more precisely which in turn will increase the performance of the whole approach. Moreover, we use a simpler network for the decoder compared to the encoder. From our experiment we saw that asymmetric architectures for encoder, decoder, and discriminator allow the network to achieve better results in both domains.

5. CONCLUSIONS

The results of evaluating each network on source sounds of each machine type and machine ID are represented in Table 1 and the similar results for target audio files are shown in Table 2. In this paper, we presented an unsupervised anomaly detection approach for DCASE 2021 challenge Task 2. We pro-posed a combination of GAN-based and autoencoder architecture in order to find the latent vector space of the non-anomalous audio clips and use it for detecting the anomalous audio files. The method has shown a better or similar results rather than baseline methods in terms of AUC specifically on the target data from the shifted domain which means that the proposed method is robust to domain shifts and can achieve acceptable performance in the cases where the audios are recorded from different environments.

6. ACKNOWLEDGEMENT

The authors wish to acknowledge CSC-IT Center for Science, Finland, for computational resources. Part of the work leading to this report has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 957337, project MARVEL

7. REFERENCES

[1] T. Tagawa, Y. Tadokoro, and T. Yairi, "Structured denoising autoencoder for fault detection and analysis," in *Proceedings of the Sixth Asian Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Phung and H. Li, Eds., vol. 39. PMLR, Nov 2015, pp. 96–111.

[2] O. I. Provotar, . M. Linder, and M. M. Veres, "Unsupervised anomaly detection in time series using lstm-based autoencoders," in *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, 2019, pp. 513–517.

[3] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, "A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 1996–2000.

[4] S. K. Lim, Y. Loo, N.-T. Tran, N.-M. Cheung, G. Roig, and Y. Elovici, "Doping: Generative data augmentation for unsupervised anomaly detection with gan," in *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 1122–1127.

[5] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," in *2018 Wireless Telecommunications Symposium (WTS)*, 2018, pp. 1–5.

[6] N. Merrill and A. Eskandarian, "Modified autoencoder training and scoring for robust unsupervised anomaly detection in deep learning," *IEEE Access*, vol. 8, pp. 101 824–101 833, 2020.

[7] W. Jiang, C. Cheng, B. Zhou, G. Ma, and Y. Yuan, "A novel gan-based fault diagnosis approach for imbalanced industrial time series," 2019.

[8] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80, Jul. 2018, pp. 4393–4402.

[9] Y. Zhou, X. Liang, W. Zhang, L. Zhang, and X. Song, "Vae-based deep svdd for anomaly detection," *Neurocomputing*, vol. 453, pp. 131–140, 2021.

[10] G. M. and U. S., "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS ONE*, vol. 11, Apr. 2016.

[11] I. Syarif, A. Prugel-Bennett, and G. Wills, "Unsupervised clustering approach for network anomaly detection," in *Fourth International Conference on Networked Digital Technologies (NDT)*, vol. 293, Apr. 2012.

[12] A. Reddy, M. Ordway-West, M. Lee, M. Dugan, J. Whitney, R. Kahana, B. Ford, J. Muedsam, A. Henslee, and M. Rao, "Using gaussian mixture models to detect outliers in seasonal univariate network traffic," in *2017 IEEE Security and Privacy Workshops (SPW)*, 2017, pp. 229–234.

[13] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection," *ACM Computing Surveys*, vol. 54, no. 2, p. 1–38, Apr. 2021. [Online]. Available: http://dx.doi.org/10.1145/3439950

[14] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised anomaly detection via adversarial training," in *Computer Vision – ACCV 2018*, C. V. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Springer International Publishing, 2019, pp. 622–637.

[15] M. Tschannen, O. Bachem, and M. Lucic, "Recent advances in autoencoder-based representation learning," 2018.

[16] 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," 2021.

[17] 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," 2021.

[18] 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," 2021.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[20] P. Zhou, J. Feng, C. Ma, C. Xiong, S. C. H. Hoi, and W. E, "Towards theoretically understanding why sgd generalizes better than adam in deep learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 21 285–21 296.