ANOMALY DETECTION USING AUTOENCODER, IDNN AND U-NET USING ENSEMBLE

Technical Report

Jun'ya Yamashita, Ryosuke Tanaka, Keisuke Ikeda, Shiiya Aoyama, Satoru Hayamizu, Satoshi Tamura

Gifu University
Faculty of Engineering Yanagido 1-1, Gifu, Gifu 5011193, Japan
junya@asr.info.gifu-u.ac.jp

ABSTRACT

This paper presents our efforts for DCASE 2022 Challenge Task 2. We built several anomaly detectors based on AutoEncoder (AE), Interpolation Deep Neural Network (IDNN) with acoustic noise, U-Net with mask patches. Through experiments using those detection schemes as well as training and development data sets, we found the best model for each machine type is different. We further integrated anomaly scores obtained from every detectors by ensemble technique. Our results show that we could improve Area Under the Curve (AUC) scores particularly for target domains.

***Index Terms*—** autoencoder, interpolation deep neural network, U-Net, ensemble.

1. INTRODUCTION

Anomaly detection is a technique to detect abnormal data, using statistics, machine learning and Deep-Learning (DL) technology. Since there are high demands to predict or detect any failure in industrial fields, many researchers have devoted their efforts to accomplish a high-performance anomaly detection technique. This paper reports our activities to DCASE 2022 Challenge Task 2: Unsupervised Anomalous Sound Detection for Machine Condition Monitoring Applying Domain Generalization Techniques [1][2][3][4].

In the anomaly detection field, an autoencoder is often adopted. An autoencoder, that employs DL architecture, converts given data into low-dimensional representations in an encoder part, followed by reconstructing the original data from the vectors in a decoder part. In our work, we also employ several kinds of autoencoders. We further focus on the other DN models for anomaly detection: IDNN and U-Net. IDNN is often used for non-periodic data, including DCASE 2021 Task 2 [5]. U-Net is also useful in the image processing field, and acoustic anomaly detection is sometimes held using sound spectrogram images. We believe these models are also helpful in this task. We build these models only using normal data, and compute an error between given and reconstructed data as an anomaly score. Since the models cannot well reconstruct anomaly data which are not used for model training, higher error scores are observed for the anomaly data. This paper subsequently explorers how to incorporate these models with high performance. We at first measure performance in each method, then consider ensemble approaches. We simply choose the best model for each machine type according to our knowledge and preliminary experimental results. In addition, we adopt a grid-search scheme to combine our models with the best mixture.

Table 1: Anomaly detection models.

Model	Audio input representation		# model parameters
	Spectrogram	Dimension	
AE0	log-mel	(640,1)	–
Mblnt	log	(128,64)	–
AE1	log	(512,1)	3,435,489
AE2	log-mel	(512,1)	2,728,273
IDNN	log	(32,128,1)	5,602,001
U-Net	log	(64,64,1)	2,164,433

2. ANOMALY DETECTION MODEL

In order to accomplish higher anomaly detection performance in different machine types, we built several detectors and compared them. In this section, we introduce all the detection methods used in this work. Table 1 summaries all the detection schemes. And Figure 1 depicts DL modules used in our models.

2.1. Baseline AE and MobileNetV2

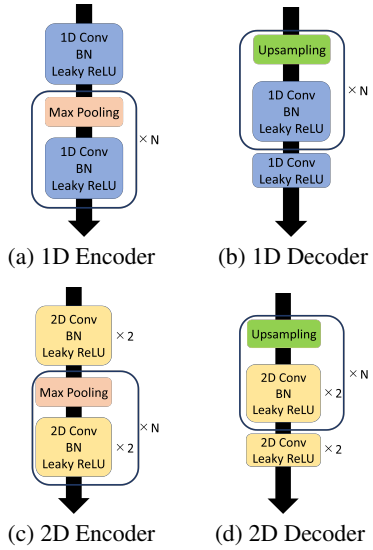
To compare our methods, we utilized both autoencoder-based and MobileNetV2-based baseline systems. In the following, the former AE model is referred as *AE0*, while *Mblnt* stands for the latter MobileNetV2 detector.

2.2. Our autoencoders

2.2.1. AE1

In addition to the baseline AE, we prepared another AE to detect anomalies. The architecture is based on Convolutional Neural Network (CNN), having the LeakyReLU as an activation function. As preprocessing, log spectrograms are calculated instead of log-mel ones. Several frequencies are combined into one bin so that the number of frequency bins could be 512, whereas the original number of bins is 128. We changed the setting from the baseline model, aiming at analyzing frequency information in detail. The standardization is then conducted to the vectors. The AE model then simply accepts the 512-dimensional single frame vector. Note that the size of the bottleneck layer is 32. Table 2 shows the model structure.

Given a vector, a reconstruction error is estimated as an anomaly score. Gaussian noise having a standard deviation of 0.01 is overlapped to testing signals, before an anomaly score is computed. This procedure is iteratively conducted with calculating the average score as the final anomaly score.



(a) 1D Encoder (b) 1D Decoder

(c) 2D Encoder (d) 2D Decoder

Figure 1: Our system modules.

2.2.2. *AE2*

In our preliminary experiments, we found a different model is needed for some machine types. An additional AE, shown in Table 3, is thus built in this work, having a 1D CNN. Different from AE1, this model uses log-mel spectraograms just like the baseline AE. We adopted 1d-CNN to observe data in the time domain in detail. While AE1 uses one-frame acoustic feature, this model requires a concatenated vector from four frames; a 512-dimensional vector is input to AE2, to estimate the same vector by the model.

The original baseline encoder is based on a Fully-Connected Neural Network (FCNN). As widely known, a CNN has a better spatial inductive bias than an FCNN [6]. Therefore, we consider CNN can produce useful representation in its middle layer.

2.3. IDNN

As an IDNN model [7] was useful in our experiments in DCASE2021, we again employed this architecture. IDNN can well capture non-periodic information, which some machines had in this task. Given a log spectrogram, a 128-dimensional vector is obtained in each frame, similar to our AE approach. Focusing on a particular frame, we concatenate vectors from its previous 16 frames and incoming 16 frames, as an input of the IDNN model. The IDNN consists of 2D-CNN-based encoder and 1D-CNN-based decoder to predict the frame [5], as Table 4. The anomaly score is obtained in the same way as our autoencoders.

2.4. U-Net

We finally chose a U-Net model [8] to estimate an anomaly score. The model has a CNN-based architecture, and accepts a 64×64 spectrogram image. To generate the image, a log spectrogram is computed with the number of frequency bins of 64 and a frame size of 64. After applying the standardization, a masked image is prepared from each spectrogram image. To obtain the image, at first the image is divided into 64 patches each having a 8×8 size. Among them, 48 patches are randomly chosen, in which the zero value is artificially filled. The U-Net is built so that the original spectrogram image could be estimated from the masked image. By introducing this masked architecture, the U-Net is expected to reconstruct normal data correctly even if only a few cues are available e.g. in noisy condition [9]. Table 5 summarizes the model.

When applying the model, the reconstruction is carried out 50 times with different masks. After that, 50 reconstructed images are averaged, followed by calculating a mean squared error between the given image and the averaged one, as an anomaly score.

Table 2: An architecture of AE1.

Input	Module	Kernel size	N	Output
(512,1)	1D Encoder	9	6	(8,512)
(8,512)	FC + BN + Leaky ReLU	-	-	32
32	FC + BN + Leaky ReLU	-	-	4096
4096	Reshape	-	-	(8,512)
(8,512)	1D Decoder	9	6	(512,1)

Table 3: An architecture of AE2.

Input	Module	Kernel size	N	Output
(512,1)	1D Encoder	5	6	(8,512)
(8,512)	FC + BN + Leahy ReLu	-	-	32
32	FC + BN + LeakyReLU	-	-	4096
4096	Reshape	-	-	(8,512)
(8,512)	1D Decoder	5	6	(512,1)

Table 4: An architecture of IDNN.

Input	Module	Kernel size	N	Output
(32,128,1)	2D Encoder	3	5	(1,4,512)
(1,4,512)	Reshape	-	-	(4,512)
(4,512)	1D Decoder	5	5	(128,1)

Table 5: An architecture of U-Net.

Input	Module	Kernel size	N	Output
(64,64,1)	2D Encoder	3	4	(4,4,256)
(4,4,256)	2D Decoder	3	4	(64,64,1)

3. SYSTEM

Using the above anomaly detection schemes, we developed four systems: two only adopting one detection method respectively, and two in ensemble manners.

3.1. System 1: IDNN

In our experiments using the development data set, IDNN performed better in three machine types: slider, valve and ToyTrain. Therefore, we employed this model as **System 1**.

3.2. System 2: U-Net

Similar to IDNN, U-Net seemed well in three machine types: slider, gearbox and ToyCar. We thus chose thie model as **System 2**.

3.3. System 3: Selective system

Experiments in the development set tell us that the best detector differs according to machine types. In order to analyze the accuracy in detail, we carried out preliminary experiments using an ensemble method. Based on the results and our knowledge, we exclusively

Table 6: Weights in System 3 (*Esmbl1*).

Machine	ToyCar	ToyTrain	bearing	fan	gearbox	slider	valve
AE1	0	0	1	1	0	0	0
IDNN	0	1	0	0	0	0	1
U-Net	1	0	0	0	1	1	0
AE2	0	0	0	0	0	0	0

Table 7: Weights in System 4 (*Esmbl2*).

machine	ToyCar	ToyTrain	bearing	fan	gearbox	slider	valve
AE1	0	0	0.827	0.127	0	0.220	0
IDNN	0.281	0.625	0.173	0.026	0	0.690	1
U-Net	0.561	0.352	0	0.146	1	0.060	0
AE2	0.138	0.023	0	0.701	0	0.030	0

Table 8: AUC [%] for source data in baseline, proposed and ensemble schemes.

Machine		AE0	Mblnt	AE1	IDNN	UNet	AE2	Esmbl1	Esmbl2
ToyCar	sec00	**86.42**	47.40	75.68	73.56	77.12	75.32	77.12	76.56
	sec01	**89.85**	62.02	78.84	66.52	79.08	84.20	79.08	74.24
	sec02	98.84	74.19	91.96	97.24	98.32	98.52	98.32	**99.12**
ToyTrain	sec00	67.54	46.02	49.52	78.12	75.84	50.88	78.12	**78.16**
	sec01	79.32	71.96	60.60	**90.64**	89.80	74.08	**90.64**	90.56
	sec02	84.08	63.23	72.08	**99.92**	98.08	81.68	**99.92**	99.88
bearing	sec00	57.48	67.85	70.40	**89.28**	86.04	61.00	70.40	71.08
	sec01	**71.03**	59.67	55.52	58.08	64.16	53.28	55.52	55.60
	sec02	42.34	**61.71**	53.64	37.44	25.48	53.76	53.64	53.24
fan	sec00	84.69	71.07	89.40	58.44	36.44	66.08	**89.40**	66.24
	sec01	71.69	**76.26**	60.60	60.56	62.84	61.88	60.60	61.68
	sec02	80.54	67.29	72.08	78.28	69.64	82.24	72.08	**91.64**
gearbox	sec00	64.63	63.54	81.84	72.12	**87.76**	69.64	**87.76**	**87.76**
	sec01	67.66	66.68	64.48	**77.12**	75.48	74.44	75.48	75.48
	sec02	75.38	80.87	64.68	77.16	**81.56**	76.76	**81.56**	**81.56**
slider	sec00	81.92	87.15	84.76	82.80	90.04	82.72	90.04	**90.12**
	sec01	67.85	49.66	71.20	89.32	**90.48**	76.08	**90.48**	85.88
	sec02	**86.66**	72.70	80.28	83.52	81.32	81.96	81.32	83.44
valve	sec00	54.24	75.26	68.76	**86.56**	65.60	60.56	**86.56**	**86.56**
	sec01	50.45	54.78	68.84	**80.72**	66.88	58.88	**80.72**	**80.72**
	sec02	51.56	76.26	63.96	**99.32**	75.40	58.84	**99.32**	**99.32**

Table 9: Acoustic feature and model training setup.

Frame length	128ms
Frame shift	64ms
# epochs	100
Learning rate	0.0001
Optimizer	Adam[10]
Batch size	128

chose one model for each machine type. The system stands for *Esmbl1*. Table 6 indicates which model is used in each machine.

3.4. System 4: Grid search

In practice, it is quite useful to utilize multiple results from different detectors. We carried out a grid search to find the best mix of the anomaly detection methods, so that the harmonic average of AUCs of source and target data as well as pAUC could be maximized. In the following, *Esmbl2* indicates this system. Table 6 represents weights for our models in each machine.

4. RESULT

We conducted experiments for sections 0, 1 and 2 in the development set, based on the setup shown in Table 9. Tables 8 and 10 show AUC results for source and target data, respectively. And Table 11 indicates pAUC results for all seven machine types. Our models could obtain better results than the baselines in most cases. IDNN and U-Net models achieved good results in the particular machines as expected. Regarding the ensemble methods, *Esmbl1* collected the best result in each machine type, furthermore, better performance was obtained in *Esmbl2*.

Comparing Tables 8 and 10, we found that the target performance slightly decreased from the source one. In terms of autoencoders, our models are superior to the baseline AE in most cases, owing to the CNN structure probably with avoiding overfitting to training data. According to Table 11, improvement of pAUC is small by using our schemes. This is because, distributions of anomaly scores of abnormal data in a source condition and data in a target condition) may be overlapped. This indicates the necessity of powerful domain adaptation technique.

Table 10: AUC [%] for target data in baseline, proposed and ensemble schemes.

Machine		AE0	Mblnt	AE1	IDNN	UNet	AE2	Esmbl1	Esmbl2
ToyCar	sec00	41.48	56.40	39.92	62.32	**69.16**	48.40	**69.16**	65.08
	sec01	41.93	56.38	53.24	**81.00**	77.68	56.24	77.68	78.92
	sec02	26.50	45.64	75.44	84.28	**84.76**	66.16	**84.76**	87.60
ToyTrain	sec00	33.68	49.41	**67.92**	67.84	59.32	55.04	67.84	66.96
	sec01	29.87	45.14	51.52	53.64	53.80	**58.44**	53.64	54.00
	sec02	15.52	44.34	64.20	81.32	**83.04**	34.16	81.32	81.84
bearing	sec00	63.07	60.17	73.32	**80.20**	76.12	58.16	73.32	73.28
	sec01	61.04	64.65	86.16	96.16	**90.08**	60.68	86.16	86.60
	sec02	52.91	**60.55**	51.00	43.04	29.72	57.12	51.00	51.08
fan	sec00	39.35	62.13	**67.92**	61.28	54.44	60.28	**67.92**	61.04
	sec01	44.74	35.12	51.52	51.80	53.92	55.96	51.52	**56.68**
	sec02	63.49	58.02	64.20	**71.44**	63.64	69.88	64.20	70.40
gearbox	sec00	64.79	67.02	73.36	72.32	**82.80**	66.72	**82.80**	**82.80**
	sec01	58.12	**66.96**	50.80	62.20	62.56	59.92	62.56	62.56
	sec02	65.57	43.15	64.60	67.60	**72.72**	65.52	**72.72**	**72.72**
slider	sec00	58.04	80.77	77.00	**79.76**	71.88	65.92	71.88	75.28
	sec01	50.30	32.07	72.48	68.84	70.56	60.80	70.56	**72.56**
	sec02	38.78	32.94	59.32	60.16	**61.32**	50.68	**61.32**	**61.32**
valve	sec00	52.73	43.60	61.64	**91.80**	46.68	49.68	**91.80**	**91.80**
	sec01	53.01	60.43	59.44	**97.72**	65.92	50.12	**97.72**	**97.72**
	sec02	43.84	**78.74**	53.96	60.44	45.44	43.80	60.44	60.44

Table 11: pAUC [%] in baseline, proposed and ensemble schemes.

Machine		AE0	Mblnt	AE1	IDNN	UNet	AE2	Esmbl1	Esmbl2
ToyCar	sec00	51.31	49.96	49.05	**53.68**	51.63	50.21	51.63	52.11
	sec01	54.08	50.92	51.11	54.47	**57.58**	51.53	**57.58**	56.68
	sec02	52.96	56.51	53.58	55.63	54.58	54.79	54.58	**62.16**
ToyTrain	sec00	52.72	50.25	50.74	**57.42**	51.74	52.74	**57.42**	56.79
	sec01	50.64	**52.97**	49.84	50.00	52.11	49.53	50.00	50.53
	sec02	48.33	51.54	52.84	55.74	**57.05**	47.89	55.74	55.89
bearing	sec00	51.49	54.41	52.89	**57.05**	55.42	50.11	52.89	52.79
	sec01	55.85	55.09	**61.53**	53.05	62.42	52.74	**61.53**	**61.53**
	sec02	49.18	**64.18**	54.00	51.26	47.47	49.68	54.00	54.00
fan	sec00	**59.95**	55.40	59.00	58.95	59.32	55.53	59.00	56.53
	sec01	51.12	**52.14**	50.42	49.74	49.89	50.58	50.42	51.05
	sec02	62.88	**65.14**	61.53	61.74	57.53	62.05	61.53	61.74
gearbox	sec00	60.93	62.12	52.05	56.42	**65.84**	64.00	**65.84**	**65.84**
	sec01	53.74	**56.85**	52.37	53.68	54.00	53.79	54.00	54.00
	sec02	61.51	50.62	59.74	60.58	**66.53**	62.47	**66.53**	**66.53**
slider	sec00	61.65	71.57	55.79	72.89	**74.58**	61.53	**74.58**	72.37
	sec01	53.06	48.21	52.00	56.79	**59.16**	54.42	**59.16**	58.37
	sec02	53.44	49.69	56.16	63.11	**64.16**	52.11	**64.16**	63.84
valve	sec00	52.15	55.37	51.58	**67.11**	50.21	51.89	**67.11**	**67.11**
	sec01	49.78	54.69	51.11	**75.47**	50.74	50.68	**75.47**	**75.47**
	sec02	49.24	**85.74**	51.26	85.26	50.68	49.79	85.26	85.26

5. REFERENCES

[1] http://dcase.community/challenge2022/task-unsupervised-detection-of-anomalous-sounds.

[2] 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, 2205.13879*, 2022.

[3] 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 Proc. of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, 2021.

[4] 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, 2206.05876*, 2022.

[5] J. Yamashita, H. Mori, S. Tamura, and S. Hayamizu, "VAE-based anomaly detection with domain adaptation." *In DCASE2021 Challenge, Tech. Rep.*, 2021.

[6] A. Ribeiro, L. Miguel Matos, P. Jose Pereira, E. C. Nunes, A. L. Ferreira, P. Cortez, and A. Pilastri. "Deep dense and convolutional autoencoders for unsupervised anomaly detection in machine condition sounds," *In arXiv, 2006.10417*, 2020.

[7] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, Y. Kawaguchi, "Anomalous Sound Detection Based on Interpolation Deep Neural Network," *In Proc. of Int'l Conf. on Acous., Speech, and Signal Process (ICASSP)*, 2020.

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *In Proc. of Med. Image Comput. Comput.-Assisted Intervention (MICCAI)*, 2015.

[9] A. Matsui, S. Asahi, S. Tamura, S. Hayamizu, R. Isashi, A. Furukawa, and T. Naitou, "Anomaly Detection in Mechanical Vibration Using Combination of Signal Processing and Autoencoder," *In J. of Signal Processing*, vol.24, no.4, pp.203-206, 2020.

[10] D. P. Kingma and J.Ba, "Adam: A method for stochastic optimization," *In Proc. of Int'l Conf. on Learning Representations (ICLR)*, 2015.