CONFORMER-BASED ID-AWARE AUTOENCODER FOR UNSUPERVISED ANOMALOUS SOUND DETECTION

Technical Report

*Tomoki Hayashi**, *Takenori Yoshimura**, *Yusuke Adachi**

Human Dataware Lab. Co., Ltd., Nagoya, Japan

{hayashi,yoshimura.takenori,adachi}@hdwlab.co.jp

ABSTRACT

This paper presents an autoencoder-based unsupervised anomalous sound detection (ASD) method for the DCASE 2020 Challenge Task 2. Inspired by the great successes of the self-attention architecture in various fields such as speech recognition, we propose Transformer- and Conformer-based autoencoder for ASD, enabling us to perform sequence-by-sequence processing. As opposed to the standard autoencoder, they can extract sequence-level information from whole audio inputs. Furthermore, we propose two simple methods for exploiting machine ID information: *machine ID embedding* and *machine ID regression*. The two methods enable the proposed models to avoid the confusion of anomalous and normal sounds among the different machine IDs. The experimental evaluation demonstrates that the proposed autoencoders outperform the conventional frame-level autoencoder, and the explicit use of machine ID information significantly improves the ASD performance. We achieved an averaged area under the curve (AUC) of 91.33% and averaged partial AUC ($p = 0.1$) of 83.34% on the development set.

***Index Terms*—** Anomaly detection, self-attention, Transformer, Conformer, autoencoder

1. INTRODUCTION

In this paper, we describe a novel autoencoder-based unsupervised ASD approach for the DCASE 2020 Challenge Task 2 [1]. Inspired by the great successes of the self-attention architecture in various fields [2–4], we propose self-attention based autoencoder architectures for unsupervised ASD. Furthermore, we propose two simple but effective methods with individual ID information to address the issue of individual differences. The contributions of this paper are summarized as follows:

- We propose a novel autoencoder architecture based on Transformer [2] and Conformer (convolution-augmented Transformer) [5]. The proposed models enable us to perform sequence-level processing to utilize the information of whole the input audio.
- We introduce two methods with the machine ID information to solve the individual difference problem: *machine ID embedding* and *machine ID regression*. These methods allow avoiding the confusion of anomalous and normal sounds among the different machine IDs.

*Equal contribution.

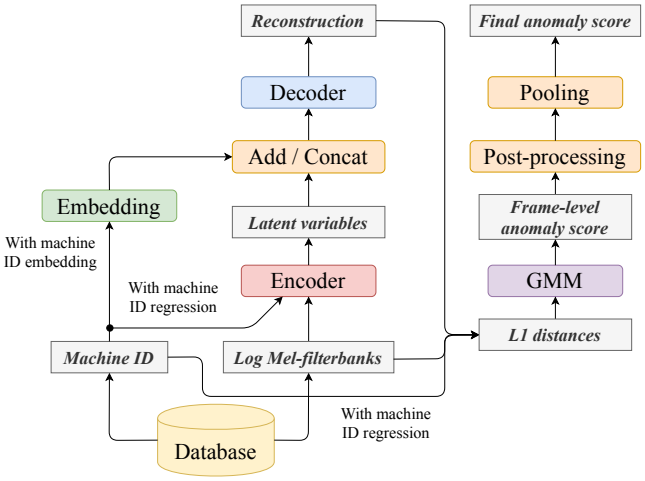

Figure 1: *Overview of the proposed method.*

- Experimental evaluation with the DCASE 2020 Challenge Task 2 dataset shows that the proposed autoencoders outperform the conventional autoencoders and that the explicit use of the machine ID significantly improves the performance. We achieve an averaged area under the curve (AUC) of 90.47% and an averaged partial AUC ($p = 0.1$) of 81.53% with a single model, and an averaged AUC of 91.33% and an averaged partial AUC of 83.34% with the ensemble.

2. PROPOSED METHOD

Figure 1 shows the overview of the proposed method. The procedure of the proposed method is described in detail in the following sections.

2.1. Audio preprocessing

First, short-term Fourier transform (STFT) of the audio waveform is computed using Hann window. The frame length and the shift length are 1,024 and 512, respectively. The STFT spectrogram is then converted to a log Mel-spectrogram with the 128 dimensional Mel basis. Finally, the log mel-spectrogram is scaled to have zero mean and unit variance using the statistics over the training data.

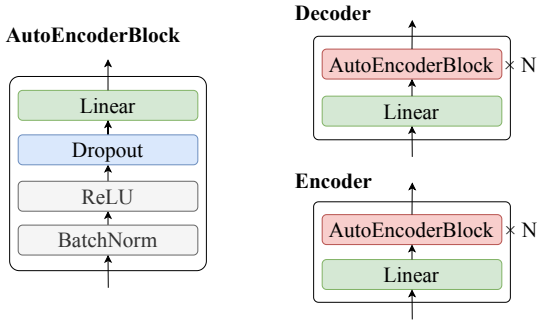

Figure 2: *Architecture of the feed-forward autoencoder.*

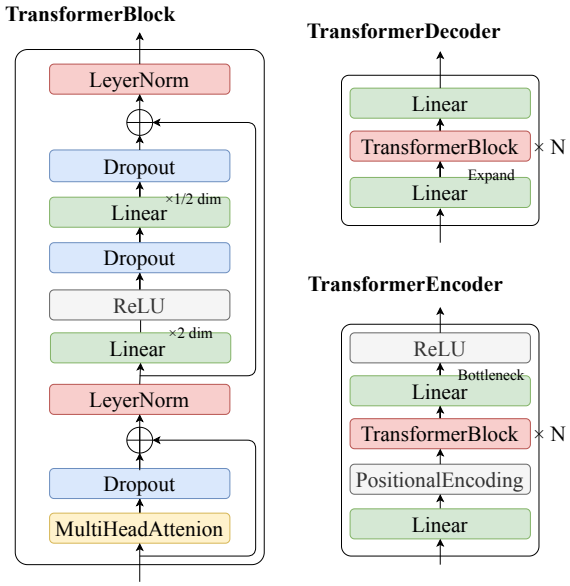

Figure 3: *Architecture of the Transformer-based autoencoder.*

2.2. Neural-network modeling

We use a simple autoencoder-based approach for unsupervised ASD. The autoencoder approach relies on the assumption that an autoencoder would not accurately reconstruct anomalous data than normal data, which is used to train the autoencoder [6]. This approach does not require a complicated training procedure such as a generative adversarial network (GAN)-based approach [7], but it can achieve comparable detection performance [8, 9].

Let us assume that the input of an autoencoder at frame t is $\boldsymbol{x}_t$ and the corresponding output is $\hat{\boldsymbol{x}}_t$. The reconstruction error $\boldsymbol{e}_t$ can be computed as follows:

$$\boldsymbol{e}_t = \mathrm{abs}(\hat{\boldsymbol{x}}_t - \boldsymbol{x}_t), \tag{1}$$

where $\mathrm{abs}(\cdot)$ denotes element-wise absolute operator. If $\boldsymbol{x}_t$ is in anomalous data, the norm of $\boldsymbol{e}_t$ should be large. Thus, anomaly detection can be performed by a simple thresholding. The architecture of the autoencoder can be designed freely.

In this study, we propose three types of autoencoder architectures: a standard feed-forward autoencoder, a Transformer-based autoencoder, and a Conformer-based autoencoder.

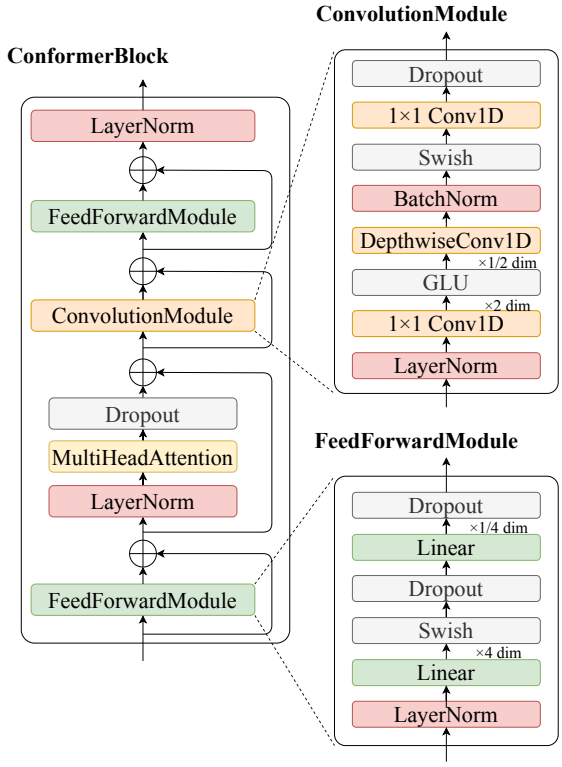

Figure 4: *Architecture of the Conformer block.*

2.2.1. Feed-forward autoencoder

Figure 2 shows the architecture of the standard feed-forward autoencoder. Both the encoder and decoder consist of a linear layer and N autoencoder blocks. Each of the autoencoder blocks consists of a 1D batch normalization layer, a rectified linear unit (ReLU) activation, a dropout layer, and a linear layer. The linear layer in the final block of the encoder has a small number of units to make a bottleneck structure. To capture the context information in the time direction, we splice five frames of the input acoustic features.

2.2.2. Transformer-based autoencoder

Transformer is a neural network with self-attention layers, achieving the great performance in various fields [2–4]. Figure 3 shows the architecture of the Transformer-based autoencoder. The encoder consists of a linear layer, a positional encoding layer, N Transformer blocks, and a linear layer with ReLU activation. The decoder is almost the same as the encoder except for the positional encoding layer and the final nonlinear activation. The structure of the Transformer block follows the original Transformer encoder architecture [2]. The final linear layer in the encoder has a small number of units as the same as the feed-forward autoencoder to make the bottleneck structure.

→ utilizzo di quest'architettura e Transformer.

2.2.3. Conformer-based autoencoder

Conformer is convolution-augmented Transformer, which achieves state-of-the-art performance in the field of automatic speech recognition (ASR) [5]. The encoder and decoder architectures of the

Conformer-based autoencoder are the same as the Transformer autoencoder as shown in Figure 3 without the positional encoding. The Transformer block is replaced with the Conformer block which is illustrated in Figure 4. The Conformer block basically consists of feed-forward modules, a multi-head attention layer, and a convolution module. The feed-forward module consists of a layer normalization layer, a linear layer with Swish activation [10] and dropout, and a linear layer with dropout. The first linear layer expands the dimension four times, and the second one projects back to the original input dimension. After both the linear layers, we multiply 0.5 to the outputs by following the method described in the original Conformer paper [5]. The convolution module consists of a layer normalization layer, a 1×1 1D convolution layer with a gated linear unit (GLU) activation [11], and a 1D depth-wise convolution layer. The depth-wise convolution layer is followed by a 1D batch normalization layer, a Swish activation, and a 1×1 1D convolution layer with dropout. We do not use the relative sinusoidal positional encoding [12] in the multi-head attention layer since the length of the inputs is fixed in the task.

2.3. ID embedding & regression

There is a possibility where normal sound of machine *A* could be anomalous sound of machine *B*. To accurately detect an anomalous sound of the target machine, we need to consider the individuality of machine, i.e., machine ID. We propose simple but effective two methods to exploiting the information: an ID embedding and an ID regression.

The ID embedding conditions the decoder with the ID embedding vector. We convert the machine ID to the embedding through an embedding layer, and the embedding is then concatenated or added to the latent variables extracted by the encoder. This makes it possible to inform the decoder of the machine ID information explicitly.

In the case of the ID regression, we simply concatenate the machine ID to the input features, and the autoencoder then reconstructs not only the input acoustic features but also the machine ID. The autoencoder tends to confuse the machine ID when the audio clip includes an anomalous sound even if we provide the correct machine ID as the inputs. Therefore, we can detect whether the audio clip includes an anomalous sound from the estimated machine ID.

2.4. GMM-based scoring

Although the frame-level reconstruction error sequence $\boldsymbol{e}_{1:T}$ can be directly used as an anomaly score with a simple averaging, we modify $\boldsymbol{e}_{1:T}$ based on the distribution of the reconstruction error to improve detection accuracy. Specifically, the frame-level anomaly score, a_t, is represented as the negative likelihood of $\boldsymbol{e}_t$ for K-mixture Gaussians:

$$a_t = -\sum_{k=1}^{K} w_k \mathcal{N}(\boldsymbol{e}_t \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{2}$$

where w_k, $\boldsymbol{\mu}_k$, and $\boldsymbol{\Sigma}_k$ are the weight, the mean vector, and the covariance matrix of the k^{th} mixture component, respectively. The parameters are trained using the reconstruction errors calculating from a validation set, which is not used for the training of the autoencoder.

2.5. Score pruning and pooling

The frame-level anomaly scores $a_{1:T}$ may contain some outliers, which could cause a negative impact on the averaging of the anomaly scores. To alleviate the impact, we remove the lowest and/or highest anomaly scores in $a_{1:T}$. The anomaly scores are sorted in ascending order:

$$\mathrm{sort}(a_{1:T}) = \left[\begin{array}{ccc} \boldsymbol{a}^{(\mathcal{L})\mathsf{T}} & \boldsymbol{a}^{(\mathcal{M})\mathsf{T}} & \boldsymbol{a}^{(\mathcal{H})\mathsf{T}} \end{array} \right], \tag{3}$$

where $\mathrm{sort}(\cdot)$ is a sorting function. The number of elements of $\boldsymbol{a}^{(\mathcal{L})}$ and $\boldsymbol{a}^{(\mathcal{H})}$ are $\lfloor Tc_1 \rfloor$ and $\lfloor Tc_2 \rfloor$, respectively, where c_1 and c_2 are in the range of $[0, 1]$. Both $\boldsymbol{a}^{(\mathcal{L})}$ and $\boldsymbol{a}^{(\mathcal{H})}$ are discarded and only $\boldsymbol{a}^{(\mathcal{M})}$ is used for averaging. Note that c_1 and c_2 are tuned so as to achieve the best detection accuracy on the development set.

As an averaging operator, the softmax weighted average is used [13]. The aggregated anomaly score $\tilde{a}$ is given by

$$\tilde{a} = \frac{1}{T'} \sum_{i=1}^{T'} a_i^{(\mathcal{M})} \left\{ \frac{\exp\left(\alpha a_i^{(\mathcal{M})}\right)}{\sum_{i'=1}^{T'} \exp\left(\alpha a_{i'}^{(\mathcal{M})}\right)} \right\}, \tag{4}$$

where $T' = T - \lfloor Tc_1 \rfloor - \lfloor Tc_2 \rfloor$ is the number of elements of $\boldsymbol{a}^{(\mathcal{M})}$ and α is a tunable scalar parameter. When $\alpha = 0$, (4) reduces to an unweighted average, and when $\alpha \to \infty$, (4) approximates max operator.

2.6. Ensemble

Combining the decisions from multiple models is often better in terms of overall performance as compared to using a single model. Let us define $\tilde{a}_g$ as an anomaly score $\tilde{a}$ predicted by a model g. The anomaly score $\tilde{a}_g$ is assumed to be normalized to $[0, 1]$ for each machine and each model. We combine the anomaly scores of different models on the basis of the performance of the models. As the performance measure, p_g, we use the average of AUC and partial AUC on the development data. The combined anomaly score $\bar{a}$ is then calculated as

$$\bar{a} = \frac{1}{G} \sum_{g=1}^{G} \tilde{a}_g \left\{ \frac{\exp\left(\beta\, p_g\right)}{\sum_{g'=1}^{G} \exp\left(\beta\, p_{g'}\right)} \right\}, \tag{5}$$

where G is the number of ensemble models and β is a tunable scalar parameter. The final anomaly score $\bar{a}$ is used for evaluation.

3. EXPERIMENTAL EVALUATION

3.1. Experimental conditions

We conducted an experimental evaluation using the DCASE 2020 Challenge Task 2 dataset [14, 15]. The dataset consisted of the normal/anomalous operating sounds of six types of toy/real machines: *Toy-car*, *Toy-conveyor*, *Fan*, *Slider*, *Pump*, *Valve*. Each recording was a single-channel, approximately 10-sec length audio sampled at 16 kHz. The number of training data was from 5,174 to 7,000 samples, and that of development data was from 879 to 3,509 samples, which depending on the machine type. The training data included only normal sounds, but the development data included both normal and anomalous sounds to check the anomaly detection performance. To verify the performance, we compared the following models:

Table 1: *Evaluation results: AUC [%] and partial AUC [%] for each machine.*

	Toy-car		Toy-conveyor		Fan		Pump		Slider		Valve		Average	
	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC
Baseline	78.77	67.58	72.53	60.43	65.83	52.45	72.89	59.99	84.76	66.53	66.28	52.54	73.51	59.92
Autoencoder	79.34	63.93	75.84	62.32	68.91	53.55	77.45	63.22	95.70	85.96	94.66	86.28	81.98	69.21
+ ID embedding	83.83	66.24	74.63	61.83	73.79	59.43	79.38	71.07	96.01	86.98	98.15	91.97	84.30	72.92
+ ID regression	82.90	69.24	73.99	61.38	72.63	64.35	79.86	72.05	95.64	85.24	96.74	88.52	83.63	73.46
Transformer	73.70	57.54	69.92	58.81	79.56	68.61	86.99	74.45	90.60	70.84	84.29	65.26	80.84	65.92
+ ID embedding	80.01	64.04	73.89	61.40	77.34	65.68	81.27	74.90	94.14	83.28	95.46	83.92	83.69	72.20
+ ID regression	84.55	73.78	74.66	62.25	85.12	73.45	88.70	**80.34**	94.34	83.06	97.71	90.11	87.51	77.16
Conformer	79.37	61.95	77.18	**63.50**	84.37	69.27	87.07	75.59	95.75	87.04	97.62	88.84	86.89	74.36
+ ID embedding	80.84	62.79	**78.01**	63.07	84.54	72.43	88.47	78.60	95.80	86.19	99.03	95.24	87.78	76.39
+ ID regression	**92.62**	**83.37**	77.91	**63.50**	**86.59**	**76.31**	**88.83**	78.43	**97.16**	**89.27**	**99.68**	**98.33**	**90.47**	**81.53**
Ensemble (Diff)	93.17	84.69	78.42	63.56	87.29	77.93	90.29	**82.08**	97.22	**90.08**	99.69	98.42	91.02	82.79
Ensemble (Best)	**93.50**	**85.44**	**79.06**	**64.75**	**87.95**	**79.30**	**90.29**	**82.08**	**97.37**	89.41	**99.82**	**99.05**	**91.33**	**83.34**

Baseline: The official baseline [1]. It was trained on the normal training data by minimizing the reconstruction error in the sense of mean squared error. The encoder consisted of four hidden layers with 128 hidden units and one hidden layer with eight hidden units. The decoder structure was symmetric of the encoder one. The batch normalization and ReLU activation units were used.

Autoencoder (ours): The proposed frame-level autoencoder. The number of linear layer units, latent variables, and layers were from 128 to 256, from 8 to 32, and from 4 to 8, respectively, which depending on the machine type. The model was trained for 100 epochs using Adam optimizer [16] with the learning rate 0.01 and the batch size 512.

Transformer (ours): The proposed Transformer-based sequence-level autoencoder. The attention dimension was 128 and the number of attention heads was four. The number of blocks and that of latent variables were from 4 to 8 and from 8 to 64, respectively, which depending on the machine type. The model was trained for 500 epochs using Noam optimizer [2] with 8,000 warmup steps, the learning rate 1.0, and the batch size 128.

Conformer (ours): The proposed Conformer-based sequence-level autoencoder. The kernel size, the attention dimension, and the number of attention heads were 31, 144, and four, respectively. The number of blocks and that of latent variables were from 4 to 8 and from 4 to 64, respectively, which depending on the machine type. The model was trained for 500 epochs using the same optimizer setting used in the Transformer autoencoder.

In addition to the above models, we also trained two variants of the above-proposed models using the proposed machine ID embedding and machine ID regression. We used 90% of training data for the autoencoder training, and the remaining 10% for the GMM training. The hyperparameters of each model and the post-processing parameters, $\{\alpha, \beta, c_1, c_2\}$, were optimized for each machine type. The evaluation metric was an area under the curve (AUC) and a partial AUC ($p = 0.1$), which were averaged over different machine IDs.

3.2. Experimental results

The experimental results are shown in Table 1. From the comparison among the different architecture models, the Transformer-based autoencoder outperformed the feed-forward autoencoder in the machine type *Fan* and *Pump*. Furthermore, the Conformer-based autoencoder outperformed the Transformer-based autoencoder and achieved the best performance in all of the machine types.

From the results using the machine ID, both the ID embedding and the ID regression improved the AUC performance for all of the proposed autoencoders. Notably, the ID regression is more effective in the case of sequence-level autoencoders. The Conformer-based autoencoder with the ID regression achieved the best performance in all of the machine types, resulting in an average AUC of 90.47% and an averaged partial AUC of 81.53%. One of the reasons is that the estimated ID in the sequence-level autoencoders tends to be consistent in the output sequence, and therefore the cumulative errors became higher compared to the frame-level autoencoder.

Finally, we performed the model ensemble. We used two strategies for the model ensemble; the ensemble of different architecture models (*Diff*) and the ensemble of 10-best models (*Best*). The result shows that both ensemble models further improved the performance and outperformed the baseline system significantly. The best ensemble model achieved an average AUC of 91.33% and averaged partial AUC of 83.34%.

4. CONCLUSION

In this paper, we presented an autoencoder-based unsupervised ASD method for DCASE 2020 Challenge Task 2. Experimental evaluation showed that the proposed Conformer-based autoencoder was effective and that using machine ID information was critical for this task. The proposed method significantly outperformed the baseline method, achieving an averaged AUC of 90.47% and an averaged partial AUC of 81.53% with a single model, and an averaged AUC of 91.33% and an averaged partial AUC of 83.34% with the ensemble. In future work, we will consider the extension of the ID regression to the classification-based approach.

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," *arXiv preprint arXiv:2006.05822*, 2020.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 2017 Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[3] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. 2019 IEEE Automatic Speech Recognition and Understanding Workshop*. IEEE, 2019, pp. 499–456.

[4] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Weakly-supervised sound event detection with self-attention," in *Proc. 2020 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2020, pp. 66–70.

[5] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[6] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *Proc. 2002 International Conference on Data Warehousing and Knowledge Discovery*, 2002, pp. 170–180.

[7] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Proc. 2018 Asian Conference on Computer Vision*. Springer, 2018, pp. 622–637.

[8] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proc. 2014 ACM Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, pp. 4–11.

[9] 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 *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015, pp. 1996–2000.

[10] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[11] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 2017 International Conference on Machine Learning*. JMLR, 2017, pp. 933–941.

[12] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[13] J. Salamon, B. McFee, P. Li, and J. P. Bello, "DCASE 2017 submission: Multiple instance learning for sound event detection," in *DCASE 2017 Challenge Technical Report*, 2017.

[14] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proc. 2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2019, pp. 308–312.

[15] 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 *Proc. the Detection and Classification of Acoustic Scenes and Events 2019 Workshop*, 2019, pp. 209–213.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.