ANOMALOUS SOUND DETECTION BY AUTO REGRESSIVE FRAME SEQUENCE MODEL

Technical Report

Yoshiharu Abe

Ralabo.jp

Yokohama, Kanagawa, Japan

ABSTRACT

The normal sound frame sequence is modeled by a base module. This base module inputs a partially masked frame sequence and predicts the masked part of the frame sequence. The anomaly score is calculated as the difference between the predicted and actual frames of the masked area. The Transformer[1] is used as the sequence model in the base module. The base module is trained with a large amount of normal sound from the source domain.

A front-end module is added in front of the base module to cope with environmental changes in the target domain. The front-end module, consisted of Transformer[1], transforms a target domain frame sequence into a source domain frame sequence. The front-end module is trained with a small amount of normal sound from the target domain.

The AUC for audio clips in the target domain was 51.11% for the domain-dependent model (with base and front-end modules), and 61.44% for the domain-independent model (with base module). Further investigation would be needed to determine why the performance of the domain-dependent model is lower than that of the domain-independent model.

Index Terms— base module, front-end module, domain adaptation, auto regressive sequence model, frame sequence, sequence to sequence, Transformer

1 INTRODUCTION

DCASE 2021 Challenge Task 2 has a large amount of normal sounds from the source domain. However, there are no anomalous sounds available for training. The sound from the machine is represented by a frame sequence of acoustic features. A base module is introduced to represent the characteristics of the frame sequences of normal sounds. The base module inputs the frame sequence x and outputs the anomaly score $A(x)$. To model the frame sequence x, Transformer[1] is used. The base module is trained with normal sounds from the source domain.

DCASE 2021 Challenge Task 2 also has a small amount of normal sounds from the target domain. In order to take advantage of the base module trained with a large amount of normal sounds in the source domain, a front-end module, which converts the sound from the target domain to the source domain, is introduced. This front-end module translates the input frame sequence from the target domain into the frame sequence of the source domain. The front-end module is trained with normal sounds from the target domain. Transformer[1] is used as the front-end module.

The calculation of anomaly score $A(x)$ is based on the auto regressive method. That is, the frame sequence x is split into two parts. One is the unmasked part u and the other is the masked part v. The base module is trained to predict the masked part v from the unmasked part u. To represent the relationship between frame positions and frame features in the frame sequence, Transformer[1] is used as the base module. In this report, for L-length frame sequence x, the first L - 1 frames is used as the unmasked part u and the last (L-th) frame is predicted as the masked part v.

2 SYSTEM OVERVIEW

Figure 1 shows the system configuration. First, the system extracts the mel spectrogram from the input sound. The system then extracts M consecutive L-frames from the mel spectrogram. Then, for each of these frame sequences, the system predicts the last frame based on the previous L-1 frame. The anomaly is based on how the last predicted frame differs from the actual frame. This can be done using the Transformer model. In this report, anomalies are measured by the mean square error between the predicted frame and the actual frame. Finally, the anomalous score of the input sound is calculated as the average of the M mean squared errors.

In the target domain, the frame sequence of the input sound is fed to the front-end module. The output of the front-end module is supplied to the base module. Both in the source and target domains the same base module ("same" means the same parameters) is used. Base module parameters are trained using the source domain sound clips. Front-end module parameters are trained using the target domain sound clips, leaving the base module parameters fixed.

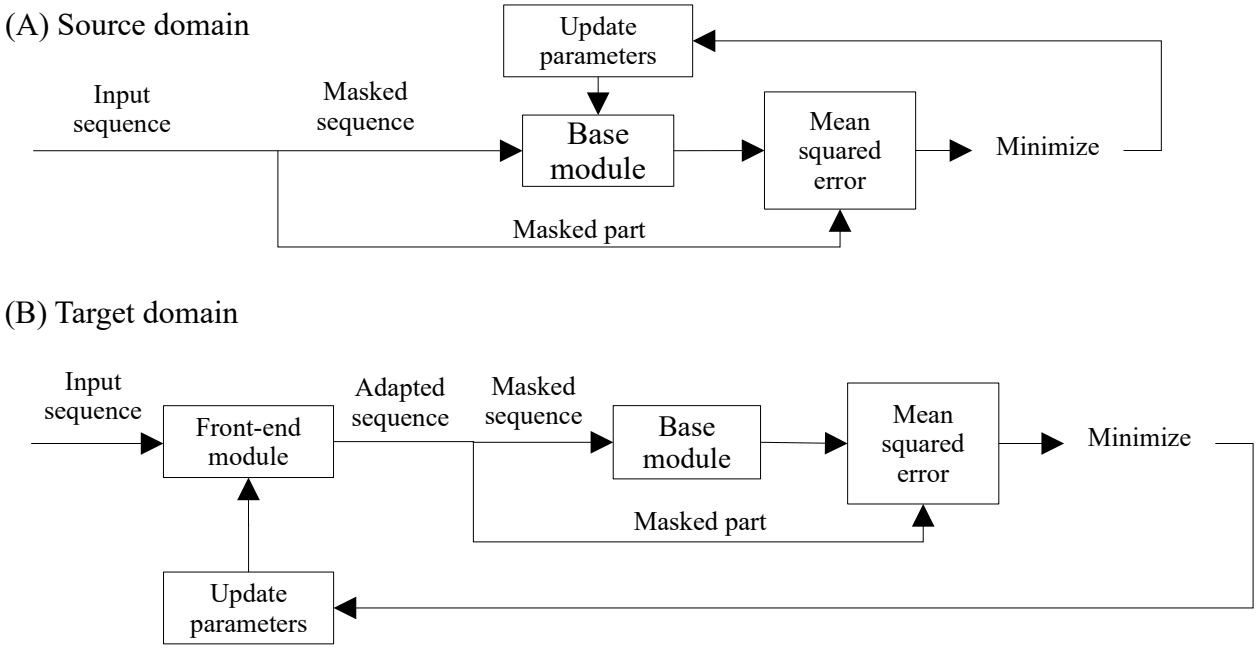

Figure 1: Flow diagram

3 BASE MODULE

The base module consists of an input layer, a position encoder, a transformer layer, and an output layer.

The input layer acts as a front end that transforms the mel spectrum frame sequence into a tensor suitable for the transformer layer. It consists of a Linear module, a LayerNorm module, a Dropout module, and a ReLU module, which inputs a tensor of size (L, D) and outputs a tensor of size (L, E). Where L is the length of the frame sequence, D is the size of the mel spectrum, and E is the embedding size (the final dimensional size of the transformer input).

The position encoder is implemented as an embedded module and its output is added to the output of the input layer.

The transformer layer is implemented using Fast Autoregressive Transformers[2]. It consists of L Transformer Encoder Layers with linear and multi-head attentions. Each TransformerEncoderLayers inputs a tensor of size (L, E) and outputs a tensor of the same size (L, E).

The output layer consists of linear modules with input sizes (L, E) and output sizes (D,). The output is used as a predicted frame and compared to the actual frame using the mean square error.

4 FRONT-END MODULE

The front-end module is used for the domain adaptation. It inputs a mel spectrum frame sequence and outputs a frame sequence utilized as the source domain frame sequence. It consists of P transformer layers implemented using Fast Auto regressive Transformers[2]. The input and output tensor size of the front-end module are both (L, D).

5 ANOMALY SCORE CALCULATION

First, the input audio clip is converted to a mel spectrum (frame) sequence, $x[0,...,T\text{-}1]$. Where T is the total length of the mel spectrum sequence. Next, from $x[0,...,T\text{-}1]$, frame sequences, $x[i,...,i+L\text{-}1]$ (i=0,...,M-1), is extracted. Where L is the length of the frame sequence and M is the number of frame sequences extracted from the audio clip. Next, the anomaly score $A[i]$ is calculated for the i-th frame sequence $x[i,...,i+L\text{-}1]$. The anomaly score $A[i]$ is calculated as the mean squared error (MSE) between the predicted frame $\tilde{x}[i+L\text{-}1]$ and the actual frame $x[i+L\text{-}1]$. The frame $\tilde{x}[i+L\text{-}1]$ is predicted from its preceding frame sequence $x[i,...,i+L\text{-}2]$ by the Transformer[1] in the base module. Finally, the anomaly score of the input audio clip is given by the average of M anomaly scores $A[i]$ (i=0,...,M-1).

Table 1: Hyper parameters

Symbols	Values	meanings
D	128	Mel-spectrum dimensions
E	256	Embedding vector dimensions
T	313	Total number of frames per sound clip
L	64	The length of the frame sequence supplied to the base module
M	250	Number of frame sequences per sound clip used to calculate anomalous scores
NH	8	Number of multi-head attentions
P	6	Number of Transformer Layers in the base module
Q	6	Number of Transformer Layers in the front-end module
BsB	32	Mini batch size for base module
BsF	8	Mini batch size for front-end module
LrB	1.0e-5	Learning rate for base module
LrF	1.0e-4	Learning rate for front-end module
NEB	8	Number of epochs for base module
NEF	64	Number of epochs for front-end module
Opt	RAdam	Parameters optimizer
FFD	1024	Feed forward dimensions
QD	256	Query dimensions(=*E*)
VD	256	Value dimensions(=*E*)
DR	0.1	Dropout rate
SR	16000	Audio sampling rate per a second
Dur	10 s	Sound clip duration
NFFT	1024	Width of frame window
NHOP	512	Frame window hopping size

Table 2: Layers in the front-end module

Layer names with arguments(names after [7])
(Front-end module)
TransformerEncoder(*layers*, norm_layer=None)
Where,
layers=[TransformerEncoderLayer(attention = 'linear', d_model = *D*, n_heads = 8, d_ff = *D**4, dropout = *DR*, activation = 'relu') for *q* =0, 1, 2, …, *Q*-1]

Table 3: Layers in the base module

Layer names with arguments(names after [6, 7])
(Input Layer)
Linear (*D*, *E*)
LayerNorm(*E*)
Dropout(*DR*)
ReLU()
(Transformer Layer)
Embedding(*L*, *E*) for positional encoding
LayerNorm(*E*)
Dropout(*DR*)
TransformerEncoder(*layers*, norm_layer=None)
Where, *layers*=[TransformerEncoderLayer(attention = 'linear', d_model = *E*, n_heads = *NH*, d_ff = *FFD*, dropout = *DR*, activation = 'relu') for *p*=0, 1, 2, …, *P*-1]
(Output Layer)
Linear(*E*, *D*)

6 EXPERIMENTS AND RESULTS

6.1 Hyper parameters

Table.1 summarizes hyper parameters in this report.

6.2 Front-end module

Table.2 shows a configuration of layers in the front-end module.

6.3 Base module

Table.3 shows a configuration of layers in the base module.

able 4: Harmonic means of AUC for test sections 00-02(in parentheses pAUC)

Models	(A) Domain-dependent model				(B) Domain-independent model trained with sections 00-02				(C) Domain-independent model trained with sections 00-05			
Modules	Base module		Base module + front-end module		Base module				Base module			
Total parameters	25.5M		37.9M		25.5M				25.5M			
Machine dependency	dependent				dependent				dependent			
Section dependency	dependent				independent				independent			
Domain dependency	dependent				independent				independent			
Data for training or adaptation	Each section's source clips		Each section's target clips		All clips of source and target in sections 00-02				All clips of source and target in sections 00-05			
Domain	Source		Target		Source		Target		Source		Target	
ToyCar	68.11	(59.64)	48.54	(48.92)	58.55	(57.24)	65.92	(57.93)	63.62	(59.85)	66.06	(57.53)
ToyTrain	66.48	(48.92)	57.03	(52.54)	69.34	(60.81)	57.57	(52.58)	66.37	(58.66)	61.50	(52.91)
Fan	63.10	(51.98)	49.81	(50.37)	65.23	(56.64)	59.42	(57.67)	56.31	(55.15)	54.07	(54.22)
Gearbox	65.36	(54.10)	45.11	(50.31)	62.61	(54.10)	70.20	(56.82)	55.80	(52.15)	66.13	(56.39)
Pump	68.27	(62.03)	51.27	(50.37)	71.77	(66.72)	61.80	(59.00)	70.18	(63.27)	56.36	(52.77)
Slider	75.90	(56.39)	55.97	(52.51)	79.18	(64.48)	67.66	(59.37)	75.94	(61.10)	64.46	(55.87)
Valve	53.60	(51.38)	51.07	(51.39)	53.56	(50.90)	49.95	(50.04)	53.63	(51.17)	52.42	(50.51)
Harmonic mean	65.52	(54.75)	51.11	(50.90)	65.27	(58.47)	61.44	(56.10)	62.66	(57.18)	59.90	(54.27)

6.4 Experimental result

Two models have been tested. One is a domain dependent model, consisted of a front-end module and a base module for each machine type and section. Only the base module is used for source domain, and the front-end module is prepended for target domain.

The other is a domain-independent model, consisted of a base module for each machine type. It was trained with clips from all sections and domains, and does not need domain adaptation.

Table 4 shows the AUC and pAUC for section 00-02 test sets as harmonic mean. As shown, the AUC for audio clips in the target domain was 51.11% for the domain-dependent model "A" (with base and front-end modules), and 61.44% for the domain-independent model "B" (with base module). Further investigation would be needed to determine why the performance of the domain-dependent model is lower than that of the domain-independent model.

Finally, for the domain-independent model, the result of increasing the amount of training data is shown. The result is shown as model "C" in Table 4. Model "C" was trained with data from section 00-05. Unfortunately, the performance was worse than the model "B". This point is also an issue for the future.

7 CONCLUSION

This report presented the auto regressive frame sequence model for unsupervised anomalous sound detection. Two types of models have been described and experimented. One is the domain-dependent model, consisted of the base module prepended by the front-end module. The other is the domain-independent model, comprised of on the base module. In the experiment, for AUC for the target domain, the performance of the domain-dependent model was lower than that of the domain-independent model. About this result further investigation would be needed.

8 REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in arXiv e-prints: 1706.03762, 2017.

[2] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention," in arXiv e-prints: 2006.16236, 2020.

[3] 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, 2021.

[4] N. Harada, D. Niizumi, D. Takeuchi, Y. Ohishi, M. Yasuda, S. Saito, "ToyADMOS2: Another Dataset of Miniature-Machine Operating Sounds for Anomalous Sound Detection under Domain Shift Conditions," in arXiv e-prints: 2106.02369, 2021.

[5] 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: 2105.02702, 2021.

[6] https://pytorch.org/, PyTorch.

[7] https://fast-transformers.github.io/transformers/, Fast Transformers for PyTorch.