Deep Convolutional Variational Autoencoder for Anomalous Sound Detection

Minh-Hieu Nguyen[1], Duy-Quang Nguyen[1], Dinh-Quoc Nguyen[1], Cong-Nguyen Pham[1], Dai Bui[2], Huy-Dung Han[1]
[1]Hanoi University of Science and Technology
[2]Confluent Inc.
Email: dung.hanhuy@hust.edu.vn

2020 IEEE Eighth International Conference on Communications and Electronics (ICCE) | 978-1-7281-5471-8/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCE48956.2021.9352085

***Abstract*—Anomalous sound detection (ASD) is one of the most important fields in industrial facility maintenance. For this task, semi-supervised approaches are preferred thanks to their simplicity and no training data labels required. These methods train an autoencoder (AE) with only normal sound data and detect anomalies based on anomaly scores of actual samples. In this paper, we propose applying the convolutional variational autoencoder (CVAE) to ASD task. Through experiments using machine sound data, the CVAE is proven to be effective in detecting abnormal sound and outperform existing methods.**

***Index Terms*—anomalous sound detection, machine sound monitoring, semi-supervised learning, autoencoder**

I. Introduction

In order to meet the demands of monitoring in a wide variety of areas, surveillance systems have been developed to automatically detect abnormal operation of the machine. Besides the commonly used security camera systems, recently audio monitoring systems have attracted a lot of attention due to their efficiency and cost-effectiveness. Anomalous sound detection (ASD) is the key technology of an audio monitoring system and has been used for various purposes including public surveillance [1], [2], animal husbandry [3], [4] and factory maintenance [5], [6]. In machine monitoring, ASD is the task to identify whether the sound emitted from a target machine is normal or anomalous [7]. The early detection of incidents based on sound can help warn of danger and thus preventing damage in machines. In this study, we focus on the abnormal machine sound detection, which contributes to the development of automated industrial monitoring systems.

The ASD problem can be solved in very simple ways such as principal component analysis (PCA) [8]. The PCA is trained to reduce the number of acoustic features by selecting the most prominent ones of normal sound. These selected components can be used for reconstructing normal sound with small difference. If the trained PCA is applied to anomalous sound, there would be a significant difference between the actual sound and the reconstructed sound, so that the anomalies can be detected. However, this method requires substantial parameter tuning for a particular application, thereby increasing the time to market.

With the emergence of deep learning in recent years, neural network-based methods have been widely applied to ASD problem [9]–[12]. These methods train an autoencoder (AE), which is the most suitable approach as it only needs normal sound to be trained. With an encoder for data compression to preserve the most important features and a decoder for original data reconstruction, the AE can detect data anomalies. During the training process, the AE updates the network parameters to minimise the anomaly score of the normal training sound data. In test scenarios, abnormal sounds can not be compressed effectively and expose high anomaly score.

AE is a very popular method for semi-supervised detection of anomalous sound and has been successfully applied to detect various types of anomalies. Some other studies utilize variational autoencoder (VAE) [9] and convolutional autoencoder (CAE) [10] to model the normal patterns. VAE is an AE-based neural network in which latent layer is represented by normal distributions. This representation can help improve the data reconstruction and avoid overfitting. Acoustic features can also be processed as image data by replacing fully-connected layers in the AE with convolutional layers, thereby constructing the CAE. In [11], ASD is considered as a statistical hypothesis test to simulate anomalous sound and the loss function of AE is defined based on the Neyman-Pearson lemma. The authors conclude that their method improves the performance measures of ASD under low false positive rate conditions. However, this approach has a drawback of using expensive rejection sampling so as to simulate anomalous sound. Wavenet [13] is another network that models the raw audio in time domain. In [12], Wavenet is utilized to operate as a predictor rather than a generator, and the anomaly score of audio data is measured by prediction error. Experimental results show that Wavenet slightly improves the efficiency of an ASD system compared to AE-based approaches. The Wavenet, however, requires a higher computational cost.

In this work, we propose applying convolutional variational autoencoder (CVAE) to ASD of industrial machine sound. Since our ASD task requires the detection of anomalies on sound segments (a segment is a combination of consecutive frames), a convolutional model like CVAE is much more suitable than fully-connected models. Furthermore, CVAE is a combination of CAE and VAE, hence the model can get the advantages of both CAE and VAE. CVAE has been used in some areas of anomaly detection, but not yet in ASD. The work in [14] deploys a sliding-window CVAE to detect anomalies of industrial robots. A fully convolutional network (FCN), a Gaussian mixture model and VAE are combined in [15] to

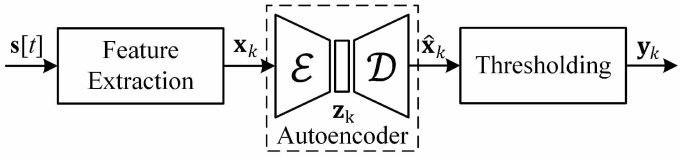


Fig. 1: ASD system model

create a Gaussian mixture fully CVAE (GMFC-VAE), which is employed for video anomaly detection. The results indicate that the CVAE outperforms traditional methods. Our study implements CVAE to detect anomalous machine sound and compare its performance to other semi-supervised methods.

II. System model

The ASD system model is illustrated in **Fig. 1**. The discrete time domain audio signal $s[t]$ is split into short-time frames (typically 25 ms) $\mathbf{s}_k$ where k is the frame index and $\mathbf{s}_k$ is the row vector of length L with the elements are taken from $s[t]$. The frequency-domain acoustic features of each frame $\mathbf{x}_k \in R^M$ is calculated by the feature extraction, where Mel filter banks (MFBs) are applied with M Mel filters. [7].

In the AE framework, the most common features of every frame are stored in the weights of the AE network during training phase. $\mathbf{x}_k$ is compressed into lower-dimensional frame $\mathbf{z}_k \in R^N$ $(N \ll M)$ by the encoder $\mathcal{E}(\cdot)$ as:

$$\mathbf{z}_k = \mathcal{E}(\mathbf{x}_k). \tag{1}$$

A reconstructed version of $\mathbf{x}_k$ can be found by the decoder $\mathcal{D}(\cdot)$ as:

$$\hat{\mathbf{x}}_k = \mathcal{D}(\mathbf{z}_k). \tag{2}$$

The AE is trained to minimise the reconstruction error (a.k.a anomaly score) of the normal training sound. The reconstruction error for each frame is defined as:

$$\mathcal{R}(\mathbf{x}_k) = \|\mathbf{x}_k - \hat{\mathbf{x}}_k\|_2^2. \tag{3}$$

During test phase, the k^{th} audio sequence is judge if it is an anomaly by means of thresholding. Let τ be the appropriate threshold, and y_k be the ASD system output, where $y_k = 1$ and $y_k = 0$ indicate that frame $\mathbf{x}_k$ is an anomalous frame and a normal frame, respectively. The anomalies are detected by:

$$y_k = \begin{cases} 1 \text{ (abnormal)} & \text{if } \mathcal{R}(\mathbf{x}_k) > \tau \\ 0 \text{ (normal)} & \text{otherwise.} \end{cases} \tag{4}$$

The optimum value of τ can be found during training phase where all $\mathbf{x}_k$ are drawn from normal audio signals.

It is noted that the sequence of audio frames is processed individually or collectively depending on the autoencoder model. In the following sections, different audio encoder models are discussed and compared.

III. Autoencoder-based approaches

A. Conventional Autoencoder

The fully-connected AE [16] is a simple network to detect anomalies. The AE has the same number of neurons in the input layer and the output layer for the purpose of input feature reconstruction. Besides, the number of nodes in neural layers is decreasing for the encoder and increasing for the decoder. The difference between the input and the output of the AE is measured by reconstruction error in **(3)**. Since the AE learns by minimizing the reconstruction error of normal training data, the actual normal data is expected to have small errors while the anomalies would have the large ones. However, due to the limited number of training data, it can not be sure that the anomaly score of actual normal sound is small for all cases. One of the reasons is that the AE processes piece-wise data without considering other data information such as distribution. Another drawback of AE is that the model is very likely to be overfitted because the objective of the AE is to reduce the reconstruction error as low as possible.

B. Variational autoencoder

To overcome the limitations of the AE, The VAE was introduced in [17]. The latent vector is represented by normal distributions instead of a neural layer. The lost function of the VAE is calculated as:

$$\mathcal{L}_{VAE} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \alpha D_{KL}, \tag{5}$$

where α is a hyperparameter and D_{KL} is the Kullback-Leibler divergence defined as:

$$D_{KL} = -\frac{1}{2}\big(1 - \mu^2 - \sigma^2 + \log(\sigma^2)\big). \tag{6}$$

The loss function of the VAE is composed of two parts: a reconstruction error as in AE and a regularisation term (D_{KL}) representing the loss of approximations of latent space distributions. D_{KL} measures the Kullback-Leibler divergence between the approximation of Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ and the standard Gaussian distribution $\mathcal{N}(0, 1)$ [17]. Here, μ and σ are the expectation and the standard deviation of the Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, respectively.

By considering the latent space distributions with D_{KL}, the VAE has a better latent space organisation than the AE. In addition, the VAE can avoid overfitting since the training is regularised. As a result, the VAE could ensure that the actual normal sound is reconstructed with small anomaly score.

C. Convolutional autoencoder

The CAE is an improved variation of the AE that sets the encoder as convolutional layers and the decoder as transposed convolutional layers [10]. Compared to the conventional AE, the CAE considers a larger scale trunk of data and, thereby, can recognize a more complex sound structure. Assuming that the audio data have many streams and each stream can be converted to K pieces of MFB of size M, The inputs to the

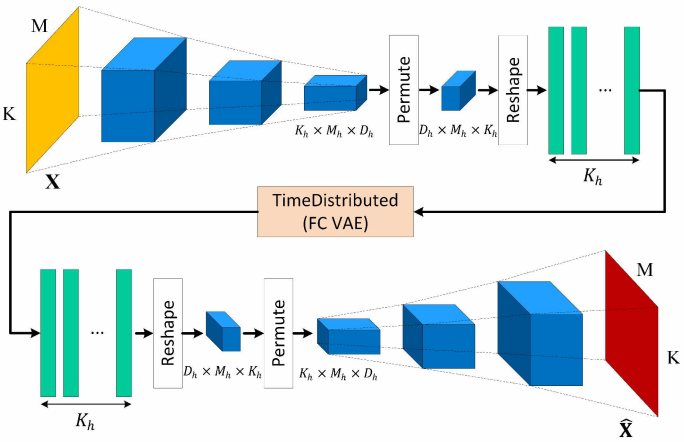


Fig. 2: Architecture of proposed CVAE

CAE can be considered as $K \times M$ images. As such, the loss function of the CAE is written as:

$$\mathcal{L}_{CAE} = \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 = \|\mathbf{X} - \mathcal{D}(\mathcal{E}(\mathbf{X}))\|_2^2, \quad (7)$$

where $\mathbf{X} = [\mathbf{x}_1^{\mathrm{T}} \ \mathbf{x}_2^{\mathrm{T}} \ \cdots \ \mathbf{x}_K^{\mathrm{T}}]^T$ is the input frequency-time image and $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1^{\mathrm{T}} \ \hat{\mathbf{x}}_2^{\mathrm{T}} \ \cdots \ \hat{\mathbf{x}}_K^{\mathrm{T}}]^T$ is the corresponding output image.

D. Convolutional variational autoencoder

The CVAE [14] is a recent improvement model of AE for more complex applications where the signal features are spread over time and frequency domain. The CVAE architecture is illustrated in **Fig. 2**. It is constructed by placing a fully-connected VAE (FC VAE), between the encoder and the decoder of the CAE. The input image $\mathbf{X}$ is fed into the convolutional layers of the encoder and produce output of dimensions K_h, M_h, D_h, where K_h, M_h and D_h represent the number of time frames, the number of acoustic features and the number of layer channels, respectively ($K_h < K$, $M_h < M$). After being permuted and reshaped, the returned data frames have dimension of K_h, $M_h \times D_h$. The TimeDistributed wrapper applies the FC VAE to every frame. The output of the TimeDistributed wrapper is reshaped, permuted and then fed into the decoder of the CVAE to return estimated image $\hat{\mathbf{X}}$.

The CVAE loss function consists of two parts: the reconstruction error as calculated in **(7)** and the Kullback-Leibler divergence D_{KL} similar to **(6)**:

$$\mathcal{L}_{CVAE} = \|\mathbf{X} - \hat{\mathbf{X}}\|_2^2 + \alpha D_{KL}, \quad (8)$$

where α is a hyperparameter and D_{KL} can be written as:

$$D_{KL} = -\frac{1}{2NK_h} \sum_{p=1}^{K_h} \sum_{q=1}^{N} (1 - \mu_{pq}^2 - \sigma_{pq}^2 + \log(\sigma_{pq}^2)), \quad (9)$$

where μ_{pq} and σ_{pq} are respectively the expectation and the standard deviation of the q^{th} Gaussian distribution in the p^{th} time frame out of K_h frames. Because the CVAE is able to learn the statistics of many frames, it is expected to captures complex features of sound.

IV. Experiments and results

A. Dataset

In our experiments, the MIMII dataset is used [18]. This dataset comprises normal and anomalous sound segments from four industrial machine types: fan, pump, slider and valve. Each machine type has four individual machines, numbered ID 00, ID 02, ID 04 and ID 06. The sampling rate of all recording sounds is 16 kHz and each sound segment is 10-second long. In addition, there are various anomalous sound scenarios are recorded, such as contamination, leakage, clogging, voltage change, rotating unbalance, rail damage, etc. Furthermore, different levels of signal-to-noise ratio (SNR) of factory noise are also considered, including -6 dB, 0 dB and 6 dB.

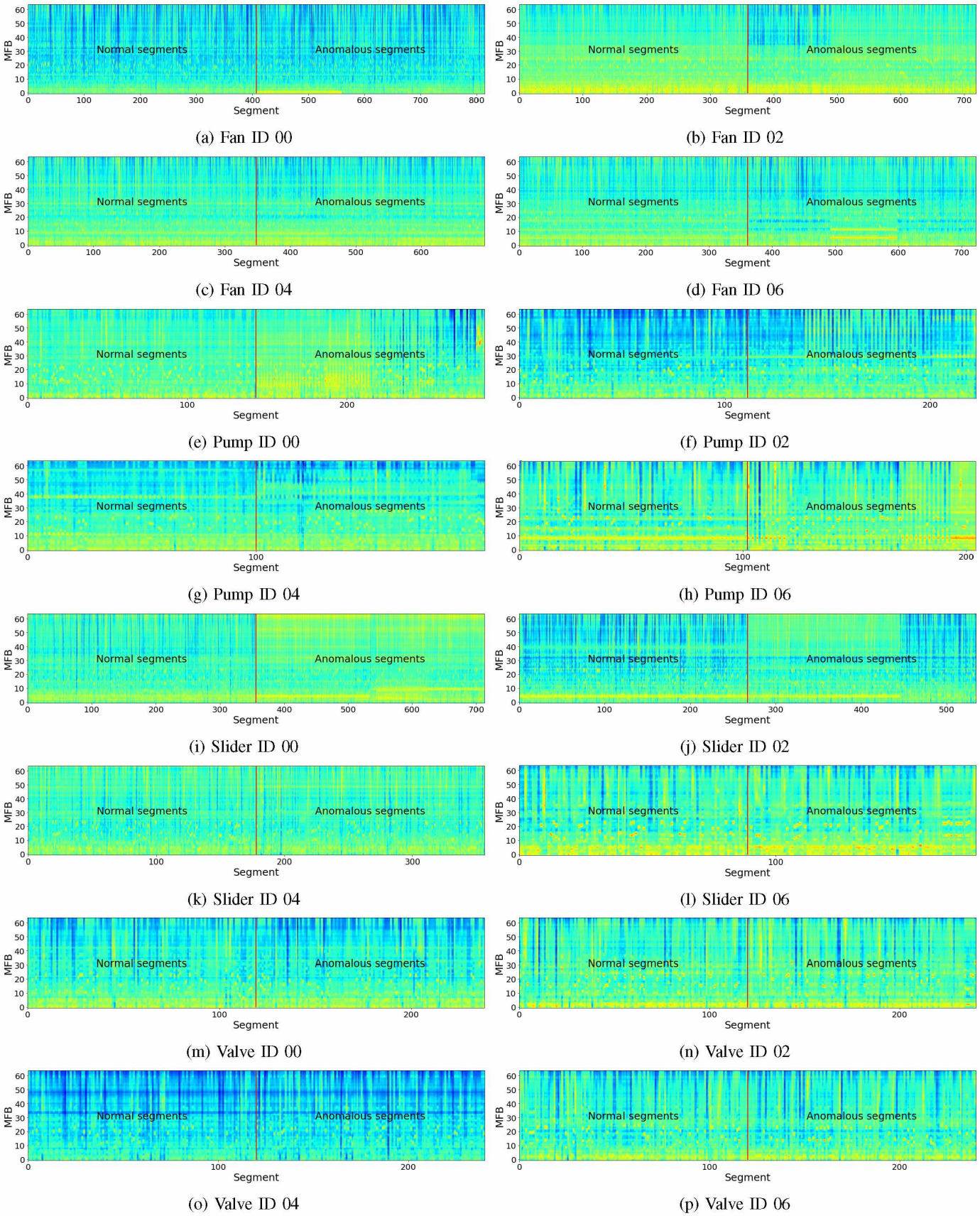
B. Experimental setup

Data for each machine from the dataset is split into a training set and a test set. The test set is composed of all anomalous segments and the same number of normal segments. The rest of normal segments are regarded as the training set. In order to extract MFB frames, we set the the number of samples between successive frames (a.k.a hop length) to 512 (note that $\frac{512}{16000} = 0.032 \sim 32$ ms) and the number of Mel filters M to 64. Thus, the inputs to the AE and the VAE are 64-dimensional vectors. For the CAE and the CVAE, since the recorded audio has a sample width of 2-byte, the number of frames for each sound segment (10 seconds) is calculated as $\lceil \frac{16000.10.2}{1024} \rceil = 313$, where $\lceil \cdot \rceil$ is the ceiling function. Therefore, the inputs to the CAE and the CVAE are 313×64 images.

The setup for four networks is summarised as follows. The AE comprises fully-connected (FC) layers: FC(32), FC(16), FC(8), FC(4), FC(2), FC(4), FC(8), FC(16), FC(32), FC(64). The setup for the VAE is the same as the one for AE, except that the FC(2) is replaced by two Gaussian approximations. The CAE encoder and decoder have three consecutive convolutional layers Conv2D(32) and three consecutive transposed convolutional layers Conv2DTransposed(32) without padding, respectively. The CVAE encoder comprises three consecutive Conv2D(32) layers and a Conv2D(1) layer with padding. The FC VAE in the CVAE includes FC(32), FC(16), two Gaussian approximations, FC(16), FC(32) and FC(64). The CVAE decoder comprises three consecutive Conv2DTransposed(32) layers and a Conv2DTransposed(1) layer. All convolutional and transposed convolutional layers use a kernel size of 3×3 and a stride size of 1×1. The hyperparameter α in **(5)** and **(8)** are empirically set to 0.01 and 0.001, respectively. The ReLU activation function [19] is used in all layers of four networks, except for the output layer. The models are trained with Adam optimizer [20] and the batch size is set to 128.

C. Metrics

The ASD system output is evaluated with segment-based evaluation using two following metrics: the area under the receiver operating characteristic (ROC) curve (AUC) and the partial-AUC (pAUC) [7]. Those metrics show the performance of a classification model at all classification threshold. Let N_n



(a) Fan ID 00 (b) Fan ID 02

(c) Fan ID 04 (d) Fan ID 06

(e) Pump ID 00 (f) Pump ID 02

(g) Pump ID 04 (h) Pump ID 06

(i) Slider ID 00 (j) Slider ID 02

(k) Slider ID 04 (l) Slider ID 06

(m) Valve ID 00 (n) Valve ID 02

(o) Valve ID 04 (p) Valve ID 06

Fig. 3: MFB spectrograms of test samples for all machines at -6 dB SNR

TABLE I: AUC results for all machines

Input SNR		-6 dB				0 dB				6 dB			
Machine	ID	AE	VAE	CAE	CVAE	AE	VAE	CAE	CVAE	AE	VAE	CAE	CVAE
Fan	00	0.5629	**0.5676**	0.5462	0.5349	0.6414	**0.6771**	0.5062	0.6115	**0.8559**	0.8285	0.6773	0.7313
	02	**0.7182**	0.7054	0.6341	0.6754	0.9266	**0.9297**	0.7762	0.8143	0.9924	**0.9949**	0.9322	0.9844
	04	0.5957	**0.7054**	0.5839	0.5405	**0.8327**	0.7942	0.7373	0.7452	0.9635	**0.9671**	0.8190	0.8949
	06	0.8723	**0.8802**	0.7452	0.7899	0.9932	**0.9943**	0.9230	0.9799	**1.0000**	**1.0000**	0.9912	0.9994
Pump	00	0.7425	0.7289	0.7574	**0.7959**	0.7646	0.7740	0.7335	**0.8252**	0.9516	0.9227	0.8522	**0.9983**
	02	0.6220	0.6080	**0.7329**	0.6276	0.6176	0.6289	**0.6790**	0.6366	0.5816	0.5991	**0.6007**	0.5578
	04	0.9215	0.9168	0.8457	**0.9369**	0.9696	0.9699	0.9076	**0.9743**	0.9991	**1.0000**	0.9937	**1.0000**
	06	0.5539	0.5849	0.6092	**0.6180**	**0.9127**	0.8945	0.7636	0.7892	0.9423	0.9856	0.9846	**0.9926**
Slider	00	0.9613	0.9573	0.9704	**0.9740**	0.9905	0.9893	0.9958	**0.9987**	0.9938	0.9945	0.9996	**0.9996**
	02	0.7514	0.7062	0.6785	**0.7770**	0.8186	0.8106	0.8236	**0.8294**	0.8908	0.9084	0.7936	**0.9166**
	04	**0.6838**	0.6627	0.5610	0.5948	0.8105	0.7851	0.8110	**0.8438**	**0.9203**	0.9163	0.7410	0.8593
	06	0.5403	0.4955	**0.6362**	0.5215	0.5200	0.5395	**0.6805**	0.5016	0.8066	0.7702	**0.8472**	0.8021
Valve	00	0.4529	0.4511	0.5015	**0.5132**	0.4943	0.4880	0.3996	**0.5153**	0.5248	0.5087	0.3477	**0.6584**
	02	0.5539	0.5263	0.5743	**0.5865**	0.5932	0.5903	0.5874	**0.6367**	0.6940	0.6440	0.6006	**0.7394**
	04	0.5134	0.5154	0.5235	**0.5253**	0.5569	0.5738	**0.6081**	0.5570	0.5701	0.5424	0.5410	**0.5875**
	06	0.4992	0.5013	0.5041	**0.5465**	0.5517	0.5347	0.5310	**0.5699**	0.5964	0.5982	0.5487	**0.6342**

TABLE II: pAUC results for all machines

Input SNR		-6 dB				0 dB				6 dB			
Machine	ID	AE	VAE	CAE	CVAE	AE	VAE	CAE	CVAE	AE	VAE	CAE	CVAE
Fan	00	0.5117	0.5111	0.5085	**0.5161**	0.5382	0.5449	0.4967	**0.5602**	**0.6587**	0.6398	0.5786	0.6408
	02	**0.5817**	0.5806	0.5774	0.5810	**0.7962**	0.7945	0.7313	0.7189	0.9601	**0.9730**	0.8352	0.9184
	04	0.5089	0.5059	0.4953	**0.5126**	**0.6187**	0.5762	0.5873	0.6152	0.8146	**0.8270**	0.6774	0.6842
	06	0.6797	0.6912	**0.7000**	0.6548	0.9645	**0.9700**	0.7383	0.9271	**1.0000**	**1.0000**	0.9542	0.9969
Pump	00	0.6944	0.6691	0.7733	**0.7883**	0.8336	0.8262	0.8524	**0.8729**	0.9370	0.9195	0.8969	**0.9950**
	02	0.6217	0.5996	**0.6338**	0.6135	0.6599	0.6609	0.6394	**0.6686**	0.7244	0.7384	**0.7391**	0.7062
	04	0.8026	0.7674	0.6521	**0.8116**	0.8742	0.8705	0.8042	**0.8768**	0.9953	**1.0000**	0.9668	**1.0000**
	06	0.5829	0.6365	0.6243	**0.6215**	**0.7914**	0.7785	0.7478	0.7667	0.7962	0.9653	0.9530	**0.9769**
Slider	00	0.8088	0.7937	0.9493	**0.9496**	0.9500	0.9438	0.9778	**0.9782**	0.9680	0.9714	0.9976	**0.9980**
	02	0.5717	0.5539	0.5341	**0.6461**	0.7371	0.7550	0.7679	**0.7861**	0.8411	0.8404	0.8128	**0.8427**
	04	**0.5557**	0.5469	0.5490	0.5445	0.6141	0.5855	0.6682	**0.6905**	0.6644	0.6522	0.6416	**0.6697**
	06	0.5043	0.5017	**0.5422**	0.5014	0.4881	0.5051	**0.5024**	0.4894	0.5333	0.5072	**0.5845**	0.5678
Valve	00	0.5065	0.5030	0.4835	**0.5150**	0.5010	0.5020	0.4821	**0.5052**	0.5246	0.5261	0.4831	**0.5770**
	02	0.5004	0.4971	0.5055	**0.5095**	0.4978	0.4967	0.5018	**0.5585**	0.5102	0.5062	0.5007	**0.5135**
	04	0.5017	0.5019	0.5068	**0.5071**	0.4949	0.5033	**0.5223**	0.4858	0.4960	0.4931	0.5044	**0.5067**
	06	0.5000	0.4963	0.5015	**0.5219**	0.4993	0.4985	0.5048	**0.5175**	0.5088	0.4934	0.4883	**0.5190**

and N_a be the number of normal and anomalous test segments, respectively. The AUC and the pAUC are defined as below:

$$AUC = \frac{1}{N_n N_a} \sum_{i=1}^{N_n} \sum_{j=1}^{N_a} \mathcal{H}(\mathcal{A}(\mathbf{x}_j^{\mathrm{a}}) - \mathcal{A}(\mathbf{x}_i^{\mathrm{n}})), \qquad (10)$$

$$pAUC = \frac{1}{\lfloor pN_n \rfloor N_a} \sum_{i=1}^{\lfloor pN_n \rfloor} \sum_{j=1}^{N_a} \mathcal{H}(\mathcal{A}(\mathbf{x}_j^{\mathrm{a}}) - \mathcal{A}(\mathbf{x}_i^{\mathrm{n}})), \qquad (11)$$

where $\{\mathbf{x}_i^{\mathrm{n}}\}_{i=1}^{N_n}$ and $\{\mathbf{x}_j^{\mathrm{a}}\}_{j=1}^{N_a}$ are respectively normal and anomalous test segments, $\lfloor \cdot \rfloor$ is the flooring function and $\mathcal{H}(x)$ returns 1 when $x > 0$ and 0 otherwise.

The pAUC is introduced to increase the true positive rate (TPR) under low false positive rate (FPR) conditions, hence it is calculated as the AUC over a low FPR range $[0, p]$. In this work, p is set to 0.1.

D. Results and discussions

Fig. 3 depicts a total of 16 MFB spectrograms of test samples for all machine types and machine IDs at -6 dB SNR. For each spectrogram, there is a red line separating segments of normal condition from segments of anomalous condition. It is clear that sound characteristics of all machines are different from each other. Additionally, we also make some observations. The following machine IDs, including the fan ID 00, the fan ID 04, the pump ID 06, the slider ID 04, the slider ID 06 and all valve IDs, have fairly similar MFB features of normal and abnormal segments. This makes it quite difficult to distinguish anomalies from normal samples. Conversely, the normal and abnormal segments of the fan ID 02, the fan ID 06, the pump ID 04, the slider ID 00 and the slider ID 02 have distinctly different properties. So, it is easier to detect anomalies for these machine IDs. Moreover, it can be seen that the MFBs of fans are relatively stationary over time, thus the frame-based models AE and VAE would be more suitable than segment-based models CAE and CVAE. Meanwhile, the valve sound is much more non-stationary, so that the reconstruction of the valve sound is really an awkward task.

Table I and **Table II** respectively show the experimental results of AUC and pAUC. The best results for each machine ID are highlighted in bold. In **Table I**, for the fan IDs regardless of SNR, two simple methods AE and VAE show higher AUCs compared to the CAE and the CVAE. For the remaining machine IDs, the proposed CVAE outperforms other

methods, except for some certain cases. Specifically, the CAE shows the best performance for the pump ID 02, the slider ID 06 and the 0 dB SNR valve ID 04. Meanwhile, the AE is the best performing model for the pump ID 06 at 0 dB SNR; and the slider ID 04 at -6 dB and 6 dB SNR. On the other hand, the AUC results for valves are significantly lower than other machines due to the instability of valve signals.

From the pAUC results displayed in **Table II**, it is obvious that four approaches show similar efficiencies to that of AUC results. However, there are some remarkable different points that can be observed. The AE and the VAE are no longer best effective approaches for the fans at all SNR levels. The CVAE has the highest pAUCs for the -6 dB and 0 dB SNR fan ID 00; and the -6 dB fan ID 04. Besides, the CAE shows the best pAUC for the fan ID 06 at -6 dB. Furthermore, the CVAE is also the best pAUC performing method for 0 dB SNR pump ID 02 and 6 dB SNR slider ID 04. To sum up, for a few specific machine IDs, the CVAE outperforms other models in terms of pAUC although its AUCs do not show the best performance, whose results are marked with an ellipse. The pAUC metric restricts the ASD system from giving false anomaly alerts, so it can evaluate the anomaly detection performance more effectively than AUC metric. If a model presents the best AUC, it does not mean it also provides the best pAUC.

In summary, the proposed CVAE shows the best result for almost machines. This is motivated by two main following reasons. Firstly, this ASD task is evaluated based on sound segments, not sound frames. The CVAE reconstructs input features segment by segment, hence it can discriminate between normal and anomalous segments more accurately than FC models which are only able of frame reconstruction. Secondly, the FC VAE with Gaussian approximations makes the CVAE both keep significant acoustic features and ensure that the latent space regularity is more general. Therefore, the CVAE can reduce the number of false detected anomalies, thereby improving AUC and pAUC, especially the latter metric. Fan is an exception in which sound signals are highly stationary in frames. For this machine type, AE and VAE give better performance than convolutional models CAE and CVAE.

V. Conclusion

In this paper, we examine the CVAE for the ASD task of industrial machine sound and compare its performance to three conventional methods including AE, VAE and CAE. The experimental results indicate that the proposed CVAE outperforms the rest for nearly all machines. Our results also demonstrate that for time stationary machine sound signals, the use of fully-connected models yields better performance in comparison with convolutional models. Additionally, it is still a challenge to reconstruct non-stationary sound as well as detect anomalies for this type of sound. This issue deserves more attention and more future studies.

Acknowledgment

This study was partially supported by Embedded Environment Sound Collection and Analysis project from SI Synergy Technology Co., Ltd and Tokyo Metropolitan Small and Medium Enterprise Support Center.

References

[1] P. Foggia, N. Petkov, A. Saggese, N. Strisciuglio, and M. Vento, "Audio surveillance of roads: A system for detecting anomalous sounds," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 1, pp. 279–288, 2015.

[2] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic novelty detection for acoustic surveillance under real-world conditions," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 713–719, 2011.

[3] Y. Chung, S. Oh, J. Lee, D. Park, H.-H. Chang, and S. Kim, "Automatic detection and recognition of pig wasting diseases using sound data in audio surveillance systems," *Sensors*, vol. 13, no. 10, pp. 12 929–12 942, 2013.

[4] W. Gutierrez, S. Kim, D. Kim, S. Yeon, and H. Chang, "Classification of porcine wasting diseases using sound analysis," *Asian-Australasian Journal of Animal Sciences*, vol. 23, no. 8, pp. 1096–1104, 2010.

[5] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing acoustic feature extractor for anomalous sound detection based on neyman-pearson lemma," in *2017 25th European Signal Processing Conference (EUSIPCO)*. IEEE, 2017, pp. 698–702.

[6] Y. Kawaguchi, R. Tanabe, T. Endo, K. Ichige, and K. Hamada, "Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 865–869.

[7] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda *et al.*, "Description and discussion on dcase2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," *arXiv preprint arXiv:2006.05822*, 2020.

[8] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[9] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[10] R. Müller, F. Ritz, S. Illium, and C. Linnhoff-Popien, "Acoustic anomaly detection for machine sounds based on image transfer learning," *arXiv preprint arXiv:2006.03429*, 2020.

[11] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 1, pp. 212–224, 2018.

[12] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, "Anomalous sound event detection based on wavenet," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 2494–2498.

[13] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[14] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, "Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder," *IEEE Access*, vol. 8, pp. 47 072–47 081, 2020.

[15] Y. Fan, G. Wen, D. Li, S. Qiu, M. D. Levine, and F. Xiao, "Video anomaly detection and localization via gaussian mixture fully convolutional variational autoencoder," *Computer Vision and Image Understanding*, p. 102920, 2020.

[16] P. Baldi, "Autoencoders, unsupervised learning, and deep architectures," in *Proceedings of ICML workshop on unsupervised and transfer learning*, 2012, pp. 37–49.

[17] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[18] H. Purohit, R. Tanabe, K. Ichige, T. Endo, Y. Nikaido, K. Suefusa, and Y. Kawaguchi, "Mimii dataset: Sound dataset for malfunctioning industrial machine investigation and inspection," *arXiv preprint arXiv:1909.09347*, 2019.

[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

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