ABNORMAL SOUND DETECTION SYSTEM BASED ON AUTOENCODER

Technical Report

Huitian Jiang[1], *Bo Lan*[2], *Huiyong Li*[3]

University of Electronic Science and Technology of China, Chengdu, China
[1] huitianjiang@std.uestc.edu.cn
[2] lan_bo@std.uestc.edu.cn
[3] hyli@uestc.edu.cn

ABSTRACT

This report describes our submissions with an autoencoder (AE) to solve the DCASE 2020 challenge task 2 (unsupervised detection of anomalous sounds for machine condition monitoring). Previous research results show that AE is a very effective solution to abnormal sound detection (ASD). This design continues previous research, using AE to implement unsupervised ASD. To decrease the false positive rate (FPR), the AE is trained to minimize the reconstruction error of normal sound. In addition, the design uses variational autoencoder (VAE) to generate normal sound samples. The generated sound samples are used to enhance AE's ability to reconstruct normal sound samples.

Index Terms— DCASE 2020, abnormal sound detection, deep learning, and autoencoder.

1. INTRODUCTION

Voice is one of the most important carriers of information, and it is the most straightforward way of communication. The amount of information transmitted by sound far exceeds that of text and images. At present, in all research directions of sound signals, the most popular should be the research of sound recognition technology. In particular, ASD has been used for a variety of purposes, including audio surveillance, animal husbandry, product inspection and predictive maintenance. For the last application, because abnormal sounds usually indicate that the mechanical equipment is malfunctioning. Discovering abnormalities quickly will reduce the number of defective products and prevent the spread of damage.

ASD tasks can be roughly divided into supervised ASD and unsupervised ASD. The difference lies in the definition of abnormal sound [9]. Supervised ASD detects "determined" abnormal sounds, such as gunshots or screams, which is a rare sound event detection (SED). Once anomalies have been defined, even if anomalies are rarer than normal sounds, we can collect a data set of target abnormal sounds. In contrast, we cannot intentionally damage expensive machines in a factory to obtain abnormal sound samples [10]-[12]. Meanwhile, the environment of factory machine operation is relatively complex. It is difficult to obtain a complete set of fault samples and apply supervised learning in fault recognition. Therefore, this type of task is reasonably considered as an unsupervised classification problem.

In this report, according to the requirements of DCASE 2020 challenge task 2, we present the design of an unsupervised ASD for industrial equipment to detect unknown abnormal sounds. A typical method of unsupervised ASD is to use outlier detection technology. The deviation between the normal model and the observed sound is calculated. Deviations are often referred to as "abnormal scores." The normal model represents the concept of normal behavior trained from normal sound training data. When the abnormal score is higher than a predetermined threshold, the observed sound is recognized as an abnormal sound.

2. DATASET

2.1. DCASE 2020 Task 2 Dataset

The dataset of DCASE 2020 challenge task 2 consists of 3 subsets. They are the development dataset, additional training dataset and evaluation dataset. In total, six types of machines are provided for investigation. They are the toy car, toy conveyor, valve, pump, fan and slide rail. Each recording is a single-channel audio with the length of roughly 10 seconds, which is a mixture of the target machine operating sound and environmental noise. In the data set, each machine type contains several different device sounds. We number each device, and the number is called ID. It is worth noting that the device IDs used in the development dataset and the evaluation dataset are different.

2.2. IDMT-ISA-ELECTRIC-ENGINE

The IDMT-ISA-ELECTRIC-ENGINE data set is composed of sound files of three similar units of the electric engine (2ACT motor brushless DC 42BLF01, 4000 RPM, 24VDC), used to simulate different acoustic conditions. The dataset contains three different working states: "good", "heavy load" and "broken". All sounds in the data set are mono audio with a sampling rate of 44.1kHz.

3. ARCHITECTURE

3.1. Network Architecture

In the actual factory environment, normal sound is very easy to obtain. When the AE model detects that the abnormal score of the sound to be detected is lower than the threshold, we can consider it as a normal sound model, so as to further train the parameters of the AE model [13]-[16].

In DCASE 2020 challenge task 2, given the normal operating sounds of different devices in six scenes, in view of the above, we first extract the sound features of the training data set, which are used to reconstruct the voice information. Then, the feature values are classified by Gaussian mixture model (GMM). The GMM learns the distribution law among the feature values and generate eigenvalues according to the learnt distribution law. If the eigenvalue conforms to the distribution error of the GMM, the generated feature is regarded as the eigenvalue of normal sound. By doing so, artificial normal sound samples are generated.

The AE was originally proposed by Marchi *et al.* [4]-[7]. In this report, two VAE models were trained and a two-level generating network was designed. The original dataset is used to train the first VAE model. The trained model generates a new dataset with the same size as the original dataset. The generated dataset is merged with the original dataset and used to train the second VAE model [17]-[18]. Similarly, another dataset is generated and used together with the original dataset to train the AE model.

encoder	Input 309*640
	Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu')

	Dense(16, name='z_mean')
	Dense(16, name='z_log_var')
	Lambda(sampling, output_shape=(16,), name='z')

decoder	Input(shape=(16,), name='z_sampling'))
	Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu')
	Dense(640)

Figure 2: The structure of the VAE model. Among them, the "z" layer of encoder is both the output of the encoder module and the input of the decoder module, which is the intermediate feature connecting the decoder and the encoder.

3.2. Features

The samples in the dataset of DCASE 2020 challenge task 2 are monaural. They have been recorded with a sampling rate of 16kHz. By using a 1024-point hamming window (samples with 512 hops), each sample or each channel of the preprocessed sample can generate a spectrogram. The log mel spectrogram is realized by applying the log mel filter bank to the spectrogram. There are 128 log mel filters in the filter bank, which cover the frequency range from 0 to 22.05 kHz. By subtracting the mean and dividing by the standard deviation, the log-mel spectrum is normalized.

Therefore, the output of MFCC is (313,128) feature "Mfcc_1", and every 5 frames before and after the obtained feature are correlated to obtain a new output (309,640) feature "Mfcc_2". The feature correlation here is essentially the same as the number of hops in MFCC. The specific association details are shown in Figure 1.

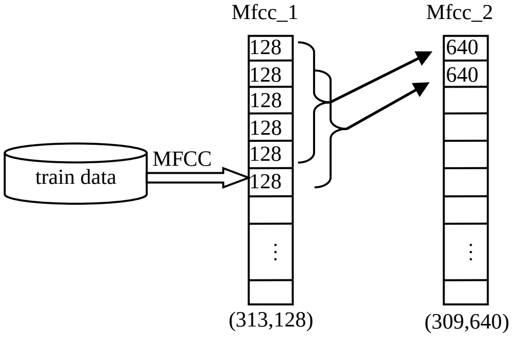

Figure 1: Feature Association Diagram

3.3. Variational Autoencoder

The first network we design for DCASE 2020 challenge task 2 is based on VAE. While training VAE, we will also train the decoder and encoder modules. The trained VAE model will be used to generate new original data. The structure of the VAE model is shown in Figure 2.

The encoder network converts the input sample x into two parameters in the hidden space, denoted as "z_mean" and "z_log_sigma". Then, we randomly sample the data point "z" from the hidden normal distribution. This hidden distribution is assumed to be the distribution that produces the input data. The calculation formula for "z" is:

$$z = z\_mean + \exp(z\_log\_sigma)*epsilon. \qquad (1)$$

In the formula, "epsilon" is a tensor that follows a normal distribution. Finally, the decoder network is used to map the hidden space to the explicit space, that is, to convert "z" back to the original input data space.

encoder	Input 309*640
	Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu')
	Dense(16)
decoder	Dense(16)
	Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu') Dense(128, activation='relu')
	Dense(640)

Figure 3: The structure of the AE model. Among them, the "Dense (16)" of the decoder and encoder represents

the same layer. The figure is only to show the structure of the decoder and the encoder separately, so they are repeated.

3.4. Gaussian Mixture Model

The second network model in this report is GMM, which analyzes the distribution of features obtained after the training data passes through the encoder. The GMM is mainly used to generate new data. We use a two-level generation networks, so two GMMs are shared in the design.

3.5. Autoencoder

The last network is the AE network, which is mainly used to calculate the abnormal score value of the input data. The design of this section imitates the structure of the VAE model. The AE model is divided into two parts, decoder and encoder, and is trained for two rounds. The structure of the AE model is shown in Figure 3. The first round uses the original data and the data generated by the first-level generation network to train the AE model. We use the original data and the encoder of the AE to train the second-level generated network model. The second round uses a mixture of original data and second-level generated network model generated data to further train the AE. Lastly, the final AE prediction model is obtained.

3.6. Network Ensemble

The use of a two-level generation network model allows the model to accurately recognize the normal sound. However, there is an uncertainty in the recognition of abnormal sounds. The training of the model by generating data through VAE reduces the generalization ability of the model to a certain extent. However, we can divide the data set by specific IDs for different scenarios, in order for the targeted training to be conducted.

We calculate the reconstruction error of the input signal as the abnormal score of the input signal. The calculation process of abnormal score is shown in Figure 4. When the score is higher than the set threshold, the input signal is regarded as an abnormal signal.

The operation of the specific system is shown in Figure 5.

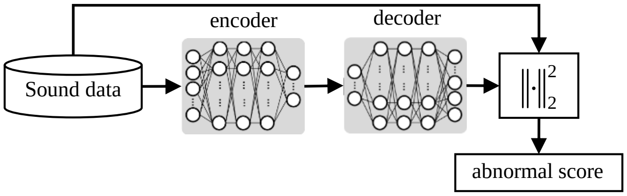

Figure 4: Calculation Process of Abnormal Score.

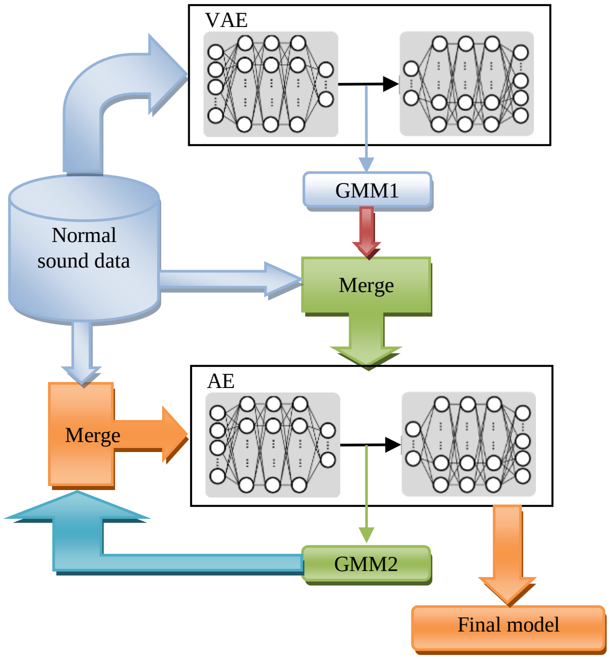

Figure 5: System Structure. It is worth noting that AE trained a total of two rounds. The input data of the first round is a mixture of original data and data generated by GMM1. The purpose is to train AE to get GMM2. The second round of training uses a mixture of raw data and data generated by GMM2. Finally, the trained parameters of the AE model are obtained.

4. RESULT

4.1. Results

The optimization function used for model training is "Adam" and the loss function is "logcosh". All parameter adjustments use default parameters.

In Table 1, we used another method to compare the results with this method. In the comparison method, we made a small improvement for the baseline. We conduct model training for each specific device. Instead of training all equipment under the same machine type. In this way, we can get a specific model for a specific device and enhance the generalization ability of the model. The results show that the method we used and the improved baseline (AE) results are better than the baseline results.

Table 1: Comparison of results of different methods.

Machine	Baseline	AE	VAE_AE
ToyCar	78.77 %	82.91%	80.12%
ToyConveyor	72.53 %	71.60%	73.06%
fan	65.83 %	71.44%	69.58%
pump	72.89 %	75.99 %	73.15%
slider	84.76 %	83.80 %	85.19%
valve	66.28 %	69.98 %	67.95%

4.2. Submissions

For final submission, the training data changed to the whole devel-opment dataset, and other configurations still to follow the local experiment., they are submitted as:
1. **Jiang_UESTC_task2_1:** This submissions is the result of using an AE model for each ID for each machine type. (AE)
2. **Jiang_UESTC_task2_2:** This submission is the result of using a combination of VAE and AE. And the GMM algorithm is used to generate new data. The amount of newly generated data is equal to the amount of original data. (VAE_AE)

5. REFERENCES

[1] Koizumi Y , Saito S , Kawachi H U Y , et al. Unsupervised Detection of Anomalous Sound based on Deep Learning and the Neyman-Pearson Lemma[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2018, PP(99).

[2] Yuma Koizumi, Shoichiro Saito, Hisashi Uematsu, Noboru Harada, and Keisuke Imoto. ToyADMOS: a dataset of miniature-machine operating sounds for anomalous sound detection. In Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 308–312.

[3] Harsh Purohit, Ryo Tanabe, Takeshi Ichige, Takashi Endo, Yuki Nikaido, Kaori Suefusa, and Yohei Kawaguchi. MIMII Dataset: sound dataset for malfunctioning industrial machine investigation and inspection. In Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019), 209–213.

[4] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, “A novel approach for automatic acoustic novelty detection using a dnoising autoencoder with bidirectional LSTM neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2015, pp. 1996–2000.

[5] T. Tagawa, Y. Tadokoro, and T. Yairi, “Structured denoising autoencoder for fault detection and analysis,” in Proc. 6th Asian Conf. Mach. Learn., 2015, pp. 96–111.

[6] E. Marchi, F. Vesperini, F. Weninger, F. Eyben, S. Squartini, and B. Schuller, “Non-linear prediction with LSTM recurrent neural networks for acoustic novelty detection,” in Proc. Int. Joint Conf. Neural Netw., 2015, pp. 1–7.

[7] Y. Kawaguchi and T. Endo, “How can we detect anomalies from subsampled audio signals?,” in Proc. IEEE 27th Int. Workshop Mach. Learn. Signal Process., 2017, pp. 1–6.

[8] Yuma Koizumi, Yohei Kawaguchi, Keisuke Imoto, Toshiki Nakamura, Yuki Nikaido, Ryo Tanabe, Harsh Purohit, Kaori Suefusa, Takashi Endo, Masahiro Yasuda, and Noboru Harada. Description and discussion on DCASE2020 challenge task2: unsupervised anomalous sound detection for machine condition monitoring. In arXiv e-prints: 2006.05822, 1–4. June 2020.

[9] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, “Scream and gunshot detection and localization for audio-surveillance systems,” in Proc. IEEE Conf. Adv. Video Signal Based Surveillance, 2007, pp. 21–26.

[10] V. J. Hodge and J. Austin, “A survey of outlier detection methodologies,” Artif. Intell. Rev., vol. 22, pp. 85–126, 2004.

[11] A. Patcha and J.-M. Park, “An overview of anomaly detection techniques: Existing solutions and latest technological trends,” J. Comput. Netw., vol. 51, pp. 3448–3470, 2007.

[12] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, 2009, Art. no. 15.

[13] E. Marchi, F. Vesperini, F. Eyben, S. Squartini, and B. Schuller, “A novel approach for automatic acoustic novelty detection using a dnoising autoencoder with bidirectional LSTM neural networks,” in Proc. IEEE Int.Conf. Acoust., Speech Signal Process., 2015, pp. 1996–2000.

[14] T. Tagawa, Y. Tadokoro, and T. Yairi, “Structured denoising autoencoder for fault detection and analysis,” in Proc. 6th Asian Conf. Mach. Learn., 2015, pp. 96–111.

[15] E. Marchi, F. Vesperini, F. Weninger, F. Eyben, S. Squartini, and B. Schuller, “Non-linear prediction with LSTM recurrent neural networks for acoustic novelty detection,” in Proc. Int. Joint Conf. Neural Netw., 2015, pp. 1–7.

[16] Y. Kawaguchi and T. Endo, “How can we detect anomalies from subsampled audio signals?,” in Proc. IEEE 27th Int. Workshop Mach. Learn. Signal Process., 2017, pp. 1–6.

[17] J. An and S. Cho, “Variational autoencoder based anomaly detection using reconstruction probability,” SNU Data Mining Center, Seoul, South Korea, Tech. Rep., 2015, pp. 1–18.

[18] Y. Kawachi, Y. Koizumi, and N. Harada, “Complementary set variational autoencoder for supervised anomaly detection,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2018, pp. 2336–2370.