DETECTION OF ANOMALOUS SOUNDS FOR MACHINE CONDITION MONITORING USING CLASSIFICATION CONFIDENCE

Tadanobu Inoue[1,†] *Phongtharin Vinayavekhin*[1,†], *Shu Morikuni*[1], *Shiqiang Wang*[2], *Tuan Hoang Trong*[2], *David Wood*[2], *Michiaki Tatsubori*[1], *Ryuki Tachibana*[1]

[1]IBM Research – Tokyo, Japan
[2]IBM Research, Yorktown Heights, NY, USA

corresponding author: pvmilk@jp.ibm.com

ABSTRACT

Anomaly-detection methods based on classification confidence are applied to the DCASE 2020 Task 2 Challenge on Unsupervised Detection of Anomalous Sounds for Machine Condition Monitoring. The final systems for submitting to the challenge are ensembles of two classification-based detectors. Both classifiers are trained with either known or generated properties of normal sounds as labels: one is a model to classify sounds into machine type and ID; the other is a model to classify transformed sounds into data-augmentation type. As for the latter model, the normal sound is augmented by using sound-transformation techniques such as pitch shifting, and data-augmentation type is used as a label. For both classifiers, classification confidence is used as the normality score for an input sample at runtime. An ensemble of these approaches is created by using probability aggregation of their anomaly scores. The experimental results on AUC show superior performance by each detector in relation to the baseline provided by the DCASE organizer. Moreover, the proposed ensemble of two detectors generally shows further improvement on the anomaly detection performance. The proposed anomaly-detection system was ranked fourth in the team ranking according to the metrics of the DCASE Challenge, and it achieves 90.93% in terms of average of AUC and pAUC scores for all the machine types, and that score is the highest of those scores achieved by all of the submitted systems.

Index Terms— Anomaly Detection, Classification-based Confident Score, Feature Learning

1. INTRODUCTION

Anomaly detection is the task of finding unusual samples in a set of data. In the setting known as "unsupervised" anomaly detection, the training data consists of only "normal" data; namely, anomalous samples are not known a priori. Algorithms for anomaly detection can be used in many applications such as quality inspection of products, maintenance of equipment, detection of network intrusions, and detection of fraud. This work focuses on detecting anomalous sounds coming from a machine to monitor the machine condition.

Anomaly detection is an important topic in machine learning. Many approaches to solve this task have been proposed [1, 2]. Deep learning is used in this work due to the type (i.e., sound) and amount of data. Approaches using deep learning for unsupervised anomaly detection can be broadly categorized as reconstruction-based methods and feature-learning-based methods. As for reconstruction-based methods, the model is trained to learn the distribution of normal samples, and anomalies can be detected by analyzing reconstruction errors as an anomaly score [3, 4, 5]. The reconstruction error is usually higher for anomalies as the model is only trained to reconstruct the normal samples. As for feature-learning-based methods, a feature-extraction model is trained to map normal data into a small region in the feature space. Anomalies can be detected by analyzing the distance from normal samples in the feature space [6, 7]. As a variation of feature-learning-based methods, classifier confidence, specifically a maximum softmax probability (MSP), can be utilized. As for this method, an anomalous sample is considered to be outside of the distributions that the classifier learned, and it generally has lower MSP [8].

To detect an anomalous sound in an unsupervised setting by using classification confidence, the following two approaches based on feature learning are taken in this study. The first approach uses normal sounds from all machine types and IDs to train a classifier that predicts the machine type and ID for each normal sound. The second approach uses data augmentation to generate pseudo classes from normal sounds. The pseudo classes are then used to learn a classifier that predicts which data-augmentation technique was used for each sound sample (generated from normal sounds). In the inference stage, a test sound clip, in which the sound is either normal or abnormal, is sent to each of these classifiers in the two approaches, to produce an anomaly score that is related to softmax probability predicted by the classifier. Finally, these anomaly scores are combined by using probability aggregation.

The two proposed classification-based approaches are described in Section 2. Techniques to improve anomaly-detection performance, including the ensemble methods, are described in Section 3. Results of experiments on the development dataset of DCASE 2020 Challenge Task 2 are presented in Section 4. Finally, the conclusions of this study are presented in Section 5.

2. FEATURE LEARNING FOR ANOMALY DETECTION

Classification-based approaches require classes in order to train a classifier to discriminate a target machine class from other machines. Two types of classifiers are trained to classify (i) machine type and ID and (ii) type of sound-data augmentation without using an external dataset. Once the anomaly score is calculated from the classifier confidence, a fixed threshold is used to determine whether a sound clip is normal or abnormal. As for unsupervised anomalous sound detection, the threshold can be defined according to the policy applied to the result of data validation, e.g., maximizing F1 score or minimizing the test escapes.

† equal contribution; alphabetical ordering

2.1. Machine type and ID as class labels

The dataset for DCASE 2020 Task 2 Challenge [9, 10, 11] has six machine types: "ToyCar," "ToyConveyor," "fan," "pump," "slider," and "valve." Each machine type has six (ToyConveyor) or seven (the other machine types) machine IDs. Tuples of machine type and ID are used as class labels. To classify a subset of these classes, a neural network is trained by using the training data in the development dataset, which contains only normal sounds. The last layer of the model is the softmax function that outputs softmax probabilities. In the inference phase, a test sound is classified by using the trained model. Since the class of each test sound, i.e., machine type and ID, is known, the anomaly score $s_1(x)$ is calculated by using the softmax probability of that particular class as follows:

$$s_1(x) = 1 - y_j(x) \tag{1}$$

where $y(x)$ is the trained model's output and j is the target-machine type and ID class index. Alternatively, MSP can also be used for the anomaly score. With this approach, none of the sound-data augmentation are used.

2.2. Types of sound-data augmentation as pseudo labels

Self-supervised feature learning [12] has been shown to be effective for tasks like anomaly detection [13, 14, 15]. It is often used when there are no labels in the training data. An anomaly-detection approach using geometric transformations on image data was proposed by Golan *et al.* [14]. With that approach, a classifier is trained to infer geometric transformations of images. The geometric transformations consist of combinations of flip, xy-translations, and rotation. During the inference phase, anomalies are detected by combining the softmax values of each geometric transformation. When these geometric transformations were naively applied to a spectrogram of sound data as images, the performance of anomaly detection was poor.

As the second approach, k types of sound-data augmentation, which includes combinations of pitch shift and time stretch, are applied to create pseudo labels to build a classifier. Then, a model is trained to classify sound segments into k classes. During the inference phase, k types of sound-data augmentation are applied to the target sound clip. The augmented k sound clips are divided into multiple sound segments and these segments are inferred by using the trained model. To get clip-wise values, the softmax probabilities are averaged over the sound segments for the target sound clip. The anomaly score $s_2(x)$ is then calculated by using the clip-wise softmax probabilities corresponding to the actual data-augmentation type as follows:

$$s_2(x) = 1 - \frac{1}{k}\sum_{j=0}^{k-1}[y(T_j(x))]_j \tag{2}$$

where $y(x)$ is the clip-wise softmax probability, $T_j(x)$ is the jth data-augmentation type, and k is the number of data-augmentation types. Anomalies can be detected on the basis of this anomaly score.

3. TECHNIQUES FOR IMPROVING PERFORMANCE

3.1. Sound segments

The proposed method can take either the whole sound clip or segments of each sound clip as classifier inputs. However, the experiments showed that segmenting a sound clip into multiple sound segments improved the performance of anomaly detection. This performance improvement might be explained by the fact that the anomaly usually occurs only within a small part of the sound clip. Each test sound is also segmented into segments, their anomaly score is individually calculated, and all the scores are aggregated by averaging.

3.2. Center loss

It has been suggested that to learn a suitable feature for one-class classification, two types of losses are required: a descriptiveness loss and a compactness loss [Perera and Patel [16]]. The classifier represents the descriptive part of the feature. How the feature is compressed is explained in the following.

An approach for anomaly detection called deep Support Vector Data Description (deep SVDD) was proposed by Ruff *et al.* [6]. With deep SVDD, a deep-learning model is trained to map normal input data to a minimized volume hypersphere in the feature space. As a result, the difference between normal and anomaly inputs in the feature space is maximized. Center loss was applied for deep face recognition by Wen *et al.* [17] to enhance the discriminative power of the deeply learned features as follows:

$$L = L_s + \lambda_c L_c \tag{3}$$

$$= L_s + \frac{\lambda_c}{2}\sum_{i=1}^{m} \| x_i - c_{y_i} \|_2^2 \tag{4}$$

where L_s is cross-entropy loss, L_c is center loss, and λ_c is center-loss weight.

In our two approaches, center loss is used for training the classifier to map normal input data to a minimized volume hypersphere in the feature space.

3.3. Ensemble methods

Ensemble methods focus on the idea of combining different results of dissimilar sub-models to enhance the overall performance of anomaly detection. Ensemble methods face challenges such as interpretability and compatibility of scores across different types of sub-models. Such challenges can be categorized as either score unification or score aggregation.

As for the proposed method, the statistical-scaling method described in [18] is used for interpreting and normalizing the scores of the sub-models. This method converts an anomaly score output by a sub-model, which cannot be directly interpreted as probability estimate into a range $[0, 1]$, e.g., autoencoder's reconstruction error-based approach. This rescaling process not only provides compatibility with other sub-models' scores for calculating later ensemble score but can also establish sufficient contrast between inliers and outliers. On the basis of the resemblance of the scores for the normal samples in the training data, the Gamma distribution was chosen under the assumption of $s_i(x) \sim \Gamma(\alpha_i, \beta_i)$, where $s_i(x)$ is the anomaly score of sub-model i. Subsequently, the scaled sub-models' scores, $\hat{s}_i(x)$, can be obtained by taking the cumulative distribution function (CDF) $F_i(x; \alpha_i, \beta_i)$ over the fitted distribution from normal samples in the training data, i.e.,

$$\begin{aligned}\hat{s}_i(x) &= F_i(x; \alpha_i, \beta_i) \\ &= \frac{\gamma_i(\alpha_i, \beta_i s_i(x))}{\Gamma(\alpha_i)}\end{aligned} \tag{5}$$

where α_i is a shape parameter and β_i is a rate parameter.

The scores of the sub-models are then combined by using probability aggregation [19] as follows:

$$f_e(x) = 1 - \prod_{i=1}^{n}(1 - f_i(x)) \tag{6}$$

where n is the number of sub-models, $f_i(x)$ is the score of each sub-model, and $f_e(x)$ is the final ensemble score. This aggregation can be used to combine either the raw anomaly score, $s_i(x)$, or the anomaly score scaled with CDF, $\hat{s}_i(x)$.

4. EXPERIMENTS

4.1. Experimental protocols

4.1.1. Audio preprocessing

In the experiment, a dataset is provided by the challenge organizer [11]. All audio clips in the dataset are sampled at 16 kHz. For each clip, short-time Fourier transform (STFT) is calculated with window size of 64 ms and hop length of 32 ms. A 128-bin logmel spectrogram is then extracted as an audio feature. For a 10 s clip, a feature is a 128 × 313 tensor. Depending on the technique used for anomaly detection, this audio feature might be divided into overlapping segments before being input into the neural network. Explicit normalization was not applied to the feature; instead, normalization layers are used in the neural network, which should provide similar normalization effect.

4.1.2. Neural-network architecture

Two neural-network architectures are used in this experiment. The first network (*inouet18*) is previously used in DCASE 2018 Task5 [20]. The architecture of the second network (*pvmilk20*) is detailed (together with number of parameters) in Table 1. Both networks use Convolutional Neural Network (CNNs), followed by global max pooling, and fully connected layers. This structure allows the models to be used with arbitrary length input.

Table 1: Information about the neural network

(a) Architecture of model *pvmilk20*.

Layer	Output size
Log Mel Spectrogram	$(ch, freq, time)$
CNN[8, k=(7, 1), s=(1, 1), p=(3, 0)] + BN + ReLU	$(8, freq, time)$
CNN[8, k=(1, 7), s=(1, 3), p=(0, 2)] + BN + ReLU	$(8, freq, time)$
CNN[32, k=(5, 1), s=(1, 1), p=(2, 0)] + BN + ReLU	$(32, freq, time)$
CNN[16, k=(5, 1), s=(1, 1), p=(2, 0)] + BN + ReLU	$(16, freq, time)$
CNN[16, k=(1, 5), s=(1, 3), p=(0, 1)] + BN + ReLU	$(16, freq, time)$
Swap(ch, freq)	$(freq, 32, time)$
Global max pooling + Dropout(0.2)	$freq$
Dense(128)	128
Dense(class) + Softmax	$class$

(b) Number of parameters of each model

Network	Parameters
inouet18 [20]	396039 to 403263
pvmilk20	22510 to 27025

Number of parameter varies with number of output $classes$.

4.1.3. Configuration for method proposed in Section 2.1

A classifier is trained by using machine type and ID as labels. The training data in the development and additional training datasets are used to train a single model to detect anomalies in the test data of both the development and evaluation datasets. No external dataset was used. A different set of hyperparameters is manually chosen to train a model for each machine type by checking the performance on the test split of the development dataset as shown in Table 2.

Table 2: Hyper-parameters for the method proposed in Section 2.1

Machine	Training data	Model	Input feature	$\boldsymbol{\lambda_c}$ [Eq. (4)]
ToyCar	*Group A*	*pvmilk20*	segment	0.1000
ToyConveyor	ToyConveyor	*pvmilk20*	segment	0.2000
fan	*Group B*	*pvmilk20*	segment	0.1000
pump	all	*pvmilk20*	clip	0.0075
slider	*Group B*	*pvmilk20*	segment	0.1000
valve	all	*inouet18*	clip	0.1500

Group A = (ToyCar, ToyConveyor) and *Group B* = (fan, pump, slider, value).

When a segment is used as an input feature, sound clips are divided into 3.072 s segments with hop length of 1.536 s. A batch size of 384 and 256 is used with the *pvmilk20* and *inouet18* models, respectively. All the models are trained by using an Adam optimizer with lr $= 0.001, \beta = (0.9, 0.999)$ and an SGD optimizer with lr $= 0.5$ on the center loss. The best model is manually chosen by considering the highest classification accuracy on the validation split in the training data.

4.1.4. Configuration for the method proposed in Section 2.2

Two sound-augmentation techniques, pitch shifting and time stretching, is used to create pseudo labels. Three different parameters are used for each technique: half steps of -0.1, 0, and $+0.1$ for pitch shifting and stretch factor of 0.9, 1.0, and 1.1 for time stretching. A total combination of 9 types of sound augmentation are thus created. These sound augmentations are then applied to sound samples of each machine ID to create pseudo labels.

For each machine type, one single classifier is trained for detecting anomalies in the test data of both the development and evaluation datasets. Pseudo labels of all machine IDs of the same machine type in the development and additional training datasets are considered as training data. That is, for all machine types except *ToyConveyor*, the classifier is trained to recognize 63 pseudo classes, which are the combination of 7 machine IDs and 9 types of sound augmentation. In the case of *ToyConveyor*, which has only 6 machines IDs, a total of $6 \times 9 = 54$ classes are available for training the classifier.

As for this second method, the *inouet18* network architecture is used for all machine types. Each sound clip is divided into 3.072 s segments with hop length of 1.536 s. Batch size is taken as 300, and center loss weight is taken as $\lambda_c = 0.005$. All the models are trained over 100 epochs using an AdamW optimizer with lr $=$ 0.001, $\beta = (0.9, 0.999)$, and weight decay $= 0.001$. An SGD optimizer is used with lr $= 0.5$ for updating center loss.

During the development of this second method, various alternative approaches to create pseudo classes have been attempted. Some examples are image transformation (flipping, xy-translations, rotation) on the logmel spectrogram, alternative parameters or more parameters for pitch shifting and time stretching, decomposing an audio sample into harmonic and percussive components, and adding harmonic and percussive sounds as pseudo classes. However, the experiments showed that the above-described configuration provided the best anomaly-detection performance.

4.1.5. Configuration for the method proposed in Section 3.3

The ensemble step consists of two main tunable parts: CDF scaling and aggregation function. For CDF scaling, two different configurations were tried to get the machine-dependent (MD) and machine-independent (MI) Gamma distributions. That is, as for the former, each machine type has its own fitted distribution for each submodel. Whereas the latter uses all six machine types together to fit one shared distribution for each sub-model. Our experimental results showed that MI is 2-7% better than MD in terms of AUC of the development test set. For the aggregation function, other than the previously mentioned method for probability estimate aggregation [19], naive arithmetic mean and product were also tested as ensemble scoring functions. While the results showed competitive AUC performance among the three tested aggregation functions, the probability aggregation function given as Equation 6 performed up to 6% better in terms of pAUC. As for these ensemble configurations, the best-performing setting, that is, MI-fitted distribution and probability estimate aggregation, was chosen for the submission.

4.2. Results and discussion

The following four types of approaches were submitted for the DCASE 2020 Task 2 Challenge:

(i) Probability aggregation [Eq. (6)]

(ii) Probability aggregation of CDF-scaled scores [Eqs. (5),(6)]

(iii) Machine types and IDs as class labels [Eq. (1)]

(iv) Sound data augmentation types as pseudo labels [Eq. (2)]

The AUC results are listed in Table 3.

Table 3: DCASE2020 Task2 Challenge performance

(a) AUC performance for development dataset

Machine type	baseline	(i)	(ii)	(iii)	(iv)
ToyCar	78.77	95.66	95.74	92.48	91.37
ToyConveyor	72.53	81.71	81.60	76.90	79.45
fan	65.83	89.05	88.73	89.13	81.27
pump	72.89	93.32	93.20	91.60	90.44
slider	84.76	99.50	99.47	99.31	98.08
valve	66.28	99.77	99.77	99.53	98.81

(b) AUC performance for evaluation dataset

Machine type	baseline	(i)	(ii)	(iii)	(iv)
ToyCar	80.14	93.80	93.16	91.06	93.06
ToyConveyor	85.36	87.32	87.41	79.88	85.82
fan	82.80	98.83	98.84	98.98	91.35
pump	82.37	94.61	94.37	93.87	92.95
slider	79.41	95.89	95.68	92.63	96.29
valve	57.37	97.69	97.82	98.02	96.07

Using a classifier trained with machine type and ID, system (iii) provides AUC $> 90\%$ for all machine types, except *ToyConveyor*. One interpretation of experimental results is the fact that the data of *ToyConveyor* are very distinct from other machine types and IDs or even among different IDs, which can be seen from the classification accuracy. This distinction makes the feature space learned from the data not so sensitive towards small changes between normal and anomaly samples. Another interesting fact is that the number of parameters of the model is relatively small, especially for the *pvmilk20* model. System (iv) using data augmentation also provides higher AUC than the baseline in the case of all machine types.

Using a sound segment as an input can improve AUC performance (2-8%) for some machine types, especially for *ToyConveyor*, i.e., *fan*, *ToyCar*, *ToyConveyor* in the case of system (iii) and *ToyConveyor* in the case of system (iv). One assumption could be that data of *ToyConveyor* has dominant background sound compared to others and processing it as a segment allows the model to recognize the background sounds and distinguish them from the anomalies.

Combining center loss with the proposed methods can improve AUC performance (2-5%) in the case of some machine types. In the challenge, both systems (iii) and (iv) use this loss in all classifiers, while the former manually chooses the weight by checking the performance obtained by the test data in the development set.

During the development, naive arithmetic mean and product were computed as ensemble scoring functions with and without CDF scaling for comparison. The ensemble with probability aggregation showed better overall performance compared to the standalone methods (see Table 3). For each machine type, the performance of the ensemble system (i) or (ii) is close to or better than that of the best standalone method, system (iii) or (iv).

On the basis of the challenge rule [11], the proposed method is 4th in team ranking, and it performs best in terms of *valve* machine type. Furthermore, it becomes 1st in the system ranking when considering the averages of AUC and pAUC over all machine types on the evaluation dataset as shown in Table 4.

Table 4: Averages of AUC and pAUC over all machine types on evaluation set for the top 10th systems in the system ranking. The proposed system (i) shows the highest average score even when all 127 systems are considered.

Team ranking	**System name (masked)**	**Average of (AUC, pAUC) on evaluation set**
1	TeamA_2_2	89.77
1	TeamA_2_1	89.74
1	TeamA_2_3	90.15
2	TeamB_2_4	90.40
3	TeamC_2_2	90.21
3	TeamC_2_1	90.59
4	**System (ii)**	90.68
4	**System (i)**	**90.93**
3	TeamC_2_4	90.03
5	TeamD_2_3	87.20

5. CONCLUSIONS

Two types of classification-based approaches are proposed to solve the anomaly-detection problem set in DCASE 2020 Task 2. The classifiers are trained using only normal sound in the development set to learn the distribution of normal data, and the model confidence is used to calculate anomaly score. The proposed systems for anomaly detection do not use an external dataset or a pre-trained model. The experimental results on AUC show superior performance on anomaly detection compared to the baseline. The proposed systems show competitive performance in the case of all machine types in the challenge with relatively small network sizes. The proposed systems are expected to improve the overall performance by incorporating approaches such as larger networks and reconstruction error-based algorithms.

6. REFERENCES

[1] C. C. Aggarwal, *Outlier Analysis*, 2nd ed. Springer Publishing Company, Incorporated, 2016.

[2] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” in *arXiv:1901.03407*, 2019.

[3] C. Zhou and R. C. Paffenroth, “Anomaly detection with robust deep autoencoders,” in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’17, New York, NY, USA, 2017, pp. 665—-674.

[4] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 146–157.

[5] D. Kimura, S. Chaudhury, M. Narita, A. Munawar, and R. Tachibana, “Adversarial discriminative attention for robust anomaly detection,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 2161–2170.

[6] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, “Deep one-class classification,” in *Proceedings of the 35th International Conference on Machine Learning*, 10–15 Jul 2018, pp. 4393–4402.

[7] P. Chong, L. Ruff, M. Kloft, and A. Binder, “Simple and effective prevention of mode collapse in deep one-class classification,” in *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2020.

[8] D. Hendrycks and K. Gimpel, “A baseline for detecting misclassified and out-of-distribution examples in neural networks,” in *Proceedings of International Conference on Learning Representations*, 2017.

[9] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. 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)*, November 2019, pp. 308–312.

[10] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2019 Workshop (DCASE2019)*, November 2019, pp. 209–213.

[11] 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,” in *arXiv e-prints: 2006.05822*, June 2020, pp. 1–4.

[12] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with convolutional neural networks,” in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 766–774.

[13] A. Munawar, P. Vinayavekhin, and G. De Magistris, “Spatio-temporal anomaly detection for industrial robots through prediction in unsupervised feature space,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 1017–1025.

[14] I. Golan and R. El-Yaniv, “Deep anomaly detection using geometric transformations,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS’18, 2018, p. 9781–9791.

[15] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, “Using self-supervised learning can improve model robustness and uncertainty,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[16] P. Perera and V. M. Patel, “Learning deep features for one-class classification,” *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5450–5463, 2019.

[17] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *Computer Vision – ECCV 2016*, 2016, pp. 499–515.

[18] H. Kriegel, P. Kröger, E. Schubert, and A. Zimek, “Interpreting and unifying outlier scores,” in *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, Dec. 2011, pp. 13–24.

[19] J. Gao and P.-N. Tan, “Converting output scores from outlier detection algorithms into probability estimates,” in *Sixth International Conference on Data Mining (ICDM’06)*, 2006, pp. 212–221.

[20] T. Inoue, P. Vinayavekhin, S. Wang, D. Wood, N. Greco, and R. Tachibana, “Domestic activities classification based on CNN using shuffling and mixing data augmentation,” DCASE2018 Challenge, Tech. Rep., September 2018.