SNIPER: FEW-SHOT LEARNING FOR ANOMALY DETECTION TO MINIMIZE FALSE-NEGATIVE RATE WITH ENSURED TRUE-POSITIVE RATE

Yuma Koizumi, Shin Murata, Noboru Harada, Shoichiro Saito, and Hisashi Uematsu

NTT Media Intelligence Laboratories, Tokyo, Japan

ABSTRACT

In anomaly detection systems, overlooking anomalies may result in serious incidents. Thus, when a system overlooks an anomaly, we need to update the system to never overlook the observed type of anomalies twice. There are roughly two possible approaches to solve this problem; re-training the whole system using all training data, or cascading a new specific detector for the overlooked anomaly. The first approach is the most effective solution; however, a huge computational cost and an amount of anomalous training data are required to re-train the system when it consists of a deep-learning-based anomaly detector. We focused on the latter approach and propose a training method for a cascaded specific anomaly detector using few-shot (just 1 to 3) samples. To suppress the false-negative rate of the overlooked anomaly, the proposed method works to decrease the false-positive rate under the constraint of true-positive rate equaling 1. Experimental results show that the proposed method outperformed conventional cross-entropy-based few-shot learning methods.

Index Terms— Few-shot learning, anomaly detection, deep learning, true-positive rate (TPR) and false-positive rate (FPR).

1. INTRODUCTION

Anomaly detection has received much attention because anomalies might indicate mistakes or malicious activities. Promptly detecting them may prevent such issues. Since microphones can capture signals without coming into contact with the monitored object, anomaly detection in sound (ADS) has been widely investigated for various purposes including surveillance [1–5], product inspection, and predictive maintenance [6–8]. For the last application, since anomalous sounds might indicate a fault in a piece of machinery, prompt detection of anomalies may prevent damage propagation. Since monitoring industrial equipment has attracted both academic and industrial attention, we take up this issue as an application of ADS.

One of the main difference between typical classification problems and ADS is the definition of the target, *i.e.,* anomalies [9]. In real-world factories, it is impractical to deliberately damage expensive machinery. In addition, actual anomalous sounds rarely occur and have high variability. Therefore, it is impossible to collect an exhaustive set of anomalous sounds and it results in anomalous sounds need to be detected for which training data does not exist. Thus, anomalies are defined as "*unknown*" sounds and detected using an outlier-detection-based unsupervised learning method [10–12] in contrast to supervised learning methods for detecting "*defined*" events such as rare sound event detection (SED) [13, 14] used in the "Detection and Classification of Acoustic Scenes and Events challenge" (DCASE) [15].

An anomaly detection system may more frequently overlook anomalies or produce false alerts with unsupervised-ADS than supervised-ADS. Overlooking anomalies may result in serious incidents. Thus, when a system overlooks an anomaly, we need to update the system using observed anomalies for it to never overlook the observed type of anomaly. There are roughly two possible approaches to achieve this; re-training the whole system using all training data, or cascading a new specific detector for the overlooked anomaly. The first approach would be impractical because retraining a deep-learning-based anomaly detector every overlooking time incurs huge computational cost. The later approach has high scalability; thus, it may be suitable in practice. However, when an observed anomaly is few-shot (just 1 to 3) audio clips, using conventional SED methods is still difficult.

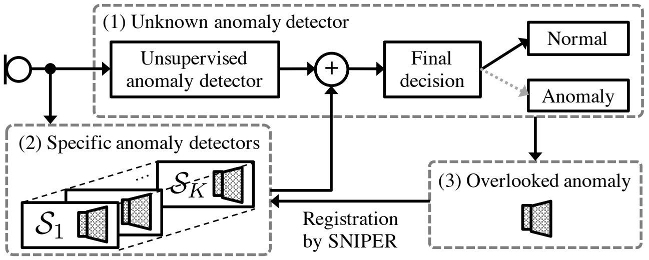

Fig. 1. Anomaly detection using (1) unsupervised anomaly detector and (2) cascaded specific anomaly detector. Overlooked anomalous sound is registered to specific anomaly detector with proposed method.

We propose a few-shot learning method for ADS to train a specific anomaly detector as shown in Fig. 1. First, we formulate the requirement of the cascaded anomaly detector, namely "*never overlook twice*". Since this requirement is satisfied when the true-positive-rate (TPR) on the observed anomaly equals 1, we build a new objective function for few-shot learning to minimize the false-positive-rate (FPR) under the TPR constraint. To calculate the TPR, we use a variational-auto-encoder (VAE) [16] for simulating other samples of the observed anomaly. We call our proposed method "few-Shot learNIng with ensured true-PositivE-Rate (SNIPER) " .

2. CONVENTIONAL METHOD

2.1. Unsupervised anomaly detection

When starting to use an ADS system to real-environments, it is often difficult to collect an exhaustive set of anomalous sounds. Thus, anomalies are defined as unknown sounds and detected using an unsupervised-one-class classifier such as outlier-detection.

In unsupervised-ADS, the deviation between a normal model and observed sound is calculated; the deviation is called the "*anomaly score*". A normal model is often constructed with a

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

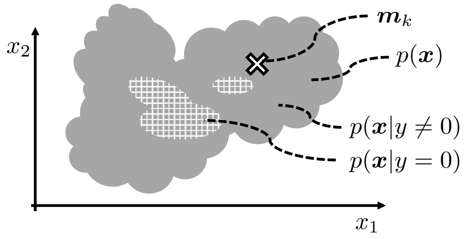

Fig. 2. PDF concept of unsupervised-ADS. PDF of normal sounds (*i.e.* meshed area) is subset of PDF of various sounds (*i.e.* whole area), and PDF of anomalous sounds is expressed as complement of normal (*i.e.* inside gray area and outside meshed area).

probability density function (PDF) of normal sounds. Accordingly, the anomaly score can be calculated as

$$\mathcal{A}(\boldsymbol{x}_t, \theta_A) = -\ln p(\boldsymbol{x}_t | y = 0, \theta_A), \tag{1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^{\mathrm{Q}}$ is an input vector calculated from the observed sound indexed t for time, θ_A is the set of parameters of the normal model, and y denotes the state; $y = 0$ is normal and $y \neq 0$ is not normal, *i.e.,* anomalous. Then, $\boldsymbol{x}_t$ is determined to be anomalous when the anomaly score exceeds a pre-defined threshold ϕ:

$$\mathcal{H}(\boldsymbol{x}_t, \phi) = \begin{cases} 0 \text{ (Normal)} & \mathcal{A}(\boldsymbol{x}_t, \theta_A) < \phi \\ 1 \text{ (Anomaly)} & \mathcal{A}(\boldsymbol{x}_t, \theta_A) \geq \phi \end{cases}. \tag{2}$$

This detection procedure implies that the universal set which consists of only the normal and anomaly, and the anomaly is defined as the complement of the normal set, as shown in Fig. 2. More intuitively, the universal set includes sounds from various machines, the normal set includes sounds from one specific machine, and anomalous sounds are all other types of machine sounds [9].

Deep learning has recently been used to construct a normal model. Several studies on deep-learning-based anomaly detection used an autoencoder (AE) [17–20], VAE [21–23] and/or normalizing flow [24]. The normal model constructed by an AE is written as

$$\mathcal{A}(\boldsymbol{x}_t, \theta_A) = \|\boldsymbol{x}_t - \mathcal{D}(\mathcal{E}(\boldsymbol{x}_t, \theta_E), \theta_D)\|^2,$$

where $\|\cdot\|$ denotes the L_2 norm, $\mathcal{E}$ and $\mathcal{D}$ are the encoder and decoder of an AE, and θ_E and θ_D are its parameters, namely $\theta_A = \{\theta_E, \theta_D\}$. Then, θ_A is trained to minimize the anomaly scores of normal sound,

$$\theta_A \leftarrow \underset{\theta_A}{\arg\min} \frac{1}{N^{(u)}} \sum_{n=1}^{N^{(u)}} \mathcal{A}(\boldsymbol{x}_n^{(u)}, \theta_A), \tag{3}$$

where $\boldsymbol{x}_n^{(u)}$ is the n-th training sample of normal sound and $N^{(u)}$ is the number of training samples of normal sound.

2.2. Specific sound event detection

While running an ADS system in a real environment, we may occasionally obtain partial samples of anomalous sounds. If the system failed to detect an anomaly, we need to update the system immediately for it to never overlook one. Unfortunately, the total number of the anomalous classes is still unknown even though the desired anomaly is defined. Therefore, since we need to update the system every time it overlooks an anomaly, the updated module requires high-scalability. Thus, we cascade specific detectors, as shown in Fig 1; each detector identifies whether the input is the desired anomaly.

There are roughly two strategies for implementation of a specific anomaly detector. An intuitive strategy is to use rare SED, which worked well in DCASE 2017 task 2 [13, 14]. However, when a new type of anomaly is detected, there is only one anomalous audio clip as the anomalous training data, thus, the use of rare SED is difficult. Another strategy for building a specific anomaly detector is using memory-based few-shot learning [25–27]. This method memorizes registered data and identifies the input data as a specific class of data when the input sample is similar to the registered data. These classifiers can be trained with few-shot (just 1 to 3) samples and a cross-entropy-like objective function. In ADS, overlooking anomalous sound is more harmful than false-alerting of normal sound. Thus, few-shot learning may be feasible, however, it may be better to modify the objective function of few-shot learning to avoid overlooking.

3. PROPOSED METHOD

3.1. SNIPER: Training policy for few-shot anomaly detection

We first define the cascaded anomaly score by adding two scores; the anomaly scores of unknown anomalous sounds $\mathcal{A}(\boldsymbol{x}_t, \theta_A)$ and the similarity scores of k-th registered anomalous sound $\boldsymbol{m}_k$ calculated by a specific anomaly detector $\mathcal{S}$ with parameter θ_S^k, as shown in Fig 1. When $K - 1$ anomalous sounds are registered, we define the cascaded anomaly score as

$$\mathcal{B}(\boldsymbol{x}, \theta_{K-1}) = \mathcal{A}(\boldsymbol{x}, \theta_A) + \gamma \sum_{k=1}^{K-1} \mathcal{S}(\boldsymbol{x}, \boldsymbol{m}_k, \theta_S^k), \tag{4}$$

where $\theta_{K-1} = \{\theta_A, \theta_S^1, ..., \theta_S^{K-1}\}$, and γ is the weight for specific anomaly detectors. When a new anomalous sound $\boldsymbol{m}_K$ is registered, $\mathcal{B}(\boldsymbol{x}, \theta_K)$ can be written as

$$\mathcal{B}(\boldsymbol{x}, \theta_K) = \mathcal{B}(\boldsymbol{x}, \theta_{K-1}) + \gamma \mathcal{S}(\boldsymbol{x}, \boldsymbol{m}_K, \theta_S^K). \tag{5}$$

Since (5) is satisfied even when $K = 1$, the problem to register an anomalous sound becomes the training of θ_S^K under given θ_{K-1}.

One performance measure of ADS consists of a TPR and FPR pair. The TPR and FPR can be calculated as expectations of $\mathcal{H}(\boldsymbol{x}, \phi)$ with respect to non-normal $p(\boldsymbol{x}|y \neq 0)$ and normal $p(\boldsymbol{x}|y = 0)$ sounds, respectively:

$$\text{TPR}(\theta_K, \phi) = \int \mathcal{H}(\boldsymbol{x}, \phi) p(\boldsymbol{x}|y \neq 0) d\boldsymbol{x}, \tag{6}$$

$$\text{FPR}(\theta_K, \phi) = \int \mathcal{H}(\boldsymbol{x}, \phi) p(\boldsymbol{x}|y = 0) d\boldsymbol{x}, \tag{7}$$

where $\mathcal{B}(\boldsymbol{x}, \theta_K)$ is used in Decision function (2) instead of $\mathcal{A}(\boldsymbol{x}, \theta_A)$. We consider $\boldsymbol{m}_K$ as a sample of $p(\boldsymbol{x}|y = K)$ and define the K-TPR as the expectation of $\mathcal{H}(\boldsymbol{x}, \phi)$ with respect to $p(\boldsymbol{x}|y = K)$ as

$$K\text{-TPR}(\theta_K, \phi) = \int \mathcal{H}(\boldsymbol{x}, \phi) p(\boldsymbol{x}|y = K) d\boldsymbol{x}. \tag{8}$$

Since overlooking anomalies may result in serious incidents, θ_K should be trained to satisfy the K-TPR(θ_K, ϕ) equaling 1. The goal with ADS is to increase the TPR and decrease FPR simultaneously; thus, we train θ_S^K to minimize the FPR under the constraint that the K-TPR(θ_K, ϕ) equals 1, *i.e.*,

$$\theta_S^K \leftarrow \underset{\theta_S^K}{\arg\min}\, \text{FPR}(\theta_K, \phi), \text{ s.t. } K\text{-TPR}(\theta_K, \phi) = 1. \tag{9}$$

This method of training θ_S^K using (9) is our proposal, *i.e.*, SNIPER.

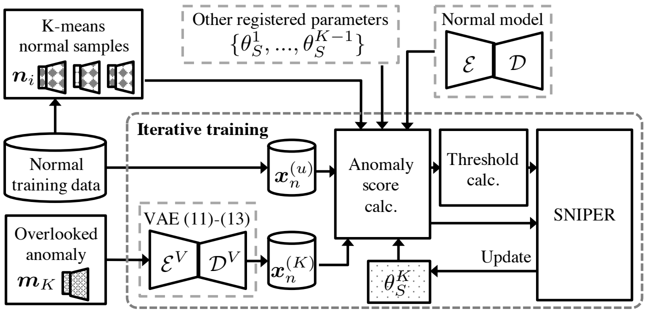

Fig. 3. A training procedure with SNIPER.

3.2. Anomalous sampling using VAE

A problem when training θ_S^K with SNIPER is the calculation of the K-TPR. The K-TPR is an expectation of $\mathcal{H}(\boldsymbol{x}, \phi)$ and is approximated as an average over the training data in most machine-learning schemes. However, in our problem scenario, we have few-shot samples; thus, the average cannot be an accurate approximation of the expectation. To calculate the TPR, we calculate $p(\boldsymbol{x}|y = K)$ using a VAE and simulate samples to calculate the average.

The VAE is used to construct the generative model of $\boldsymbol{x}$ by using an encoder $\mathcal{E}^V(\boldsymbol{x}, \theta_E^V)$ and decoder $\mathcal{D}^V(\boldsymbol{z}, \theta_D^V)$ [16]. In this model, $\mathcal{E}^V(\boldsymbol{x}, \theta_E^V)$ estimates the parameters of the Gaussian distribution of the latent vector $\boldsymbol{z}$, namely mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\nu}$. Then, $\boldsymbol{z}$ is sampled from Gaussian$(\boldsymbol{\mu}, \boldsymbol{\nu})$. Finally, $\boldsymbol{x}$ is reconstructed as $\boldsymbol{x} = \mathcal{D}^V(\boldsymbol{z}, \theta_D^V)$. One of the important properties of VAE is that the PDF of a specific sample of $\boldsymbol{x}$ can be estimated. In unsupervised-ADS, anomalous sound is the complement of normal sound, and a registered sample $\boldsymbol{m}_k$ is also a sample of $p(\boldsymbol{x})$, as shown in Fig. 2. Thus, by training $\mathcal{E}^V$ and $\mathcal{D}^V$ to construct a generative model of various sounds, $p(\boldsymbol{x}|y = K)$ and K-TPR can be calculated as

$$K\text{-TPR}(\theta_K, \phi) \approx \frac{1}{M} \sum_{n=1}^{M} \mathcal{H}\left(\mathcal{B}\left(\boldsymbol{x}_n^{(K)}, \theta_K\right), \phi\right), \tag{10}$$

where M is the batchsize and the n-th sample $\boldsymbol{x}_n^{(K)}$ is simulated by

$$\boldsymbol{x}_n^{(K)} = \mathcal{D}^V(\boldsymbol{z}_n^{(K)}, \theta_D^V), \tag{11}$$

$$\boldsymbol{z}_n^{(K)} \sim \text{Gaussian}(\boldsymbol{\mu}_K, \boldsymbol{\nu}_K), \tag{12}$$

$$\boldsymbol{\mu}_K, \boldsymbol{\nu}_K = \mathcal{E}^V(\boldsymbol{m}_K, \theta_E^V), \tag{13}$$

where $\sim$ denotes a sample generation from the right-hand PDF.

3.3. Implementation

This section describes an implementation of the training procedure of SNIPER, as shown in Fig. 3. We used a squared-error-based similarity with dimension reduction-like feature extraction as an implementation of $\mathcal{S}$. In addition, since $\boldsymbol{m}_k$ is a sample of an anomalous PDF, $\boldsymbol{m}_k$ should not be similar to normal sounds. Thus, as auxiliary information, we also use the dissimilarity between I normal samples $\{\boldsymbol{n}_i\}_{i=1}^I$ calculated using the K-means algorithm from the normal training dataset. Thus, $\mathcal{S}$ can be calculated as

$$\mathcal{S}(\boldsymbol{x}, \boldsymbol{m}_k, \theta_S^k) = \frac{1}{2}\left[\mathcal{D}(f_{\boldsymbol{x}}^k, f_{\boldsymbol{m}_k}^k) - \frac{1}{I}\sum_{i=1}^{I}\mathcal{D}(f_{\boldsymbol{x}}^k, f_{\boldsymbol{n}_i}^k) + 1\right], \tag{14}$$

$$\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) = 2 \cdot \text{sigmoid}\left(-(\boldsymbol{x} - \boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y})\right), \tag{15}$$

$$f_{\boldsymbol{x}}^k = \mathbf{W}_k\left(\boldsymbol{x} \odot \text{sigmoid}(\mathbf{g}_k)\right), \tag{16}$$

where $\odot$ is the element-wise product. Thus, the parameters of $\mathcal{S}$ are $\theta_S^k := \{\mathbf{W}_k, \mathbf{g}_k\}$, where $\mathbf{W}_k \in \mathbb{R}^{\mathsf{D} \times \mathsf{Q}}$ and $\mathbf{g}_k \in \mathbb{R}^{\mathsf{Q}}$. Since $\mathcal{D}(\boldsymbol{x}, \boldsymbol{y})$ satisfies $1 \leq \mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) \leq 2$, $0 \leq \mathcal{S}(\boldsymbol{x}, \boldsymbol{m}_k, \theta_S^k) \leq 1$ and returns a large value when $\boldsymbol{x}$ is similar to $\boldsymbol{m}_k$.

Next, to constrain K-TPR equaling 1, we determine a threshold ϕ_K that satisfies K-TPR$(\theta_K, \phi_K) = 1$. To numerically calculate ϕ_K, we use the VAE described in Sec. 3.2. Since K-TPR is approximately calculated by (10), K-TPR equaling 1 is achieved when ϕ_K is smaller than the minimum value of $\mathcal{B}\left(\boldsymbol{x}_n^{(K)}, \theta\right)$. Thus, we use ϕ_K defined as

$$\phi_K \leftarrow \min\left[\left\{\mathcal{B}\left(\boldsymbol{x}_n^{(K)}, \theta\right)\right\}_{n=1}^{M}\right]. \tag{17}$$

SNIPER (9) is then possible by minimizing FPR(θ_K, ϕ_K). However, the binary decision function $\mathcal{H}$ is not differentiable; thus, the gradient with respect to θ_S^K cannot be calculated. To analytically calculate the gradient, we approximate $\mathcal{H}$ by the sigmoid function $\sigma(\cdot)$ as follows:

$$\text{FPR}(\theta_K, \phi_K) \approx \frac{1}{M}\sum_{n=1}^{M}\sigma\left(\mathcal{B}\left(\boldsymbol{x}_n^{(u)}, \theta\right) - \phi_K\right). \tag{18}$$

The anomaly score of $\boldsymbol{m}_K$ may become small when focusing only on the FPR; thus, we train θ_K to also increase K-TPR. To calculate the gradient of K-TPR, we also approximate K-TPR using $\sigma(\cdot)$. In addition, to training θ_S^K so that $\mathcal{S}(\boldsymbol{x}_n^{(u)}, \boldsymbol{m}_K, \theta_S^K)$ come close to 0, we use the following regularization:

$$\mathcal{L} = \frac{1}{M}\sum_{n=1}^{M}\ln\mathcal{S}(\boldsymbol{x}_n^{(u)}, \boldsymbol{m}_K, \theta_S^K) - \ln\mathcal{S}(\boldsymbol{x}_n^{(K)}, \boldsymbol{m}_K, \theta_S^K). \tag{19}$$

Summarizing the above discussion, using SNIPER to train θ_S^k is possible as follows:

$$\theta_S^K \leftarrow \arg\min_{\theta_S^K} \text{FPR}(\theta_K, \phi_K) - K\text{-TPR}(\theta_K, \phi_K) + \mathcal{L}. \tag{20}$$

4. EXPERIMENTS

4.1. Experimental conditions

To investigate whether SNIPER is effective for few-shot learning for ADS, we compared it with an AE-based unsupervised-ADS method and two cross-entropy-based few-shot learning methods:

- `AE`: Vanilla unsupervised-ADS using AE.
- `COS-CE`: $\mathcal{S}$ and objective function are replaced. Distance function (15) is replaced with cosine-similarity [25, 26] $\mathcal{D}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2}\left[\frac{\boldsymbol{x}^\top \boldsymbol{y}}{\|\boldsymbol{x}\| \cdot \|\boldsymbol{y}\|} + 1\right] + 1$. We add a bias vector $\mathbf{b}_k \in \mathbb{R}^{\mathsf{D}}$ for acoustic feature calculation (16). Since the objective function used in the above study [25] is similar to

the cross-entropy used in SED tasks [13, 14], θ_S^K is trained to minimize the cross-entropy as follows:

$$\theta_S^K \leftarrow \arg\min_{\theta_S^K} \sum_{n=1}^{M} \ln\left(\frac{1}{K}\sum_{k=1}^{K} \mathcal{S}(\boldsymbol{x}_n^{(u)}, \boldsymbol{m}_k, \theta_S^k)\right) + \sum_{n=1}^{M} \ln\left(1 - \frac{1}{K}\sum_{k=1}^{K} \mathcal{S}(\boldsymbol{x}_n^{(K)}, \boldsymbol{m}_k, \theta_S^k)\right), \tag{21}$$

where $\boldsymbol{x}_n^{(K)}$ is also simulated using the VAE.

- `MSE-CE`: The objective function is replaced with (21).

The difference between the proposed method and the conventional methods is the objective function. If SNIPER outperforms both `COS-CE` and `MSE-CE`, it will be more effective than cross-entropy-based training.

We used fully-connected neural networks for the AE and VAE. The encoder and detecoder had two hidden layers with 512 hidden units, and the rectified linear unit was used for each layer. The inputs were the log amplitude spectrum of 40-dimensional Mel-filterbank outputs, and before/after 5 frames were concatenated for the AE. Thus, the dimension of the AE's input vector was 440 and that of the VAE was 40. The AE and VAE were trained iusing normal data and various data, respectively, which are described in the next section.

For the input of $\mathcal{S}$, the log amplitude spectrum of the 40-dimensional Mel-filterbank outputs with a context window of size $C = 10$ was used. Thus, the dimension of the input vector was $\mathsf{Q} = 880 (= 40 \times (2C + 1))$. The other parameters were set as $\mathsf{D} = 4$, $\gamma = 5.0$, $M = 100$, and $I = 3$. All methods were trained using the gradient descent algorithm for ten epochs.

4.2. Dataset

We used the same dataset of ADS for machines used in a previous study [9]. Sounds emitted from a condensing unit of an air conditioner operating in a real environment were used as normal sounds. Various machine sounds were also recorded from other machines, including a *compressor*, *engine*, *compression pump*, and *electric drill*, as well as the environmental noise of factories. The normal and various-machine sound data totaled 4 and 20 hours, respectively. A part of the training dataset used for the DCASE-2016 task [28, 29] was used as anomalous sounds; 140 sounds including *slamming doors* , *knocking at doors* , *keys put on a table*, *keystrokes on a keyboard*, *drawers being opened*, *pages being turned*, and *phones ringing*) were selected. To synthesize the test data, the anomalous sounds were mixed with normal sounds at anomaly-to-normal power ratios (ANRs) of -15 and -20 dB. All sounds were recorded at a 16-kHz sampling rate. The frame size and shift size of the DFT were 512 and 256 points, respectively.

4.3. Objective evaluations

We tested under $K = 1$ and 3 situations, and "*slamming doors* " and "*pages being turned*" were used as registered anomalous categories. From these test data, 21 frames including audio events were cut and used as $\boldsymbol{m}_k$. The training ANR condition of $\boldsymbol{m}_k$ was -15 dB. The test ANR conditions were -15 and -20 dB. Thus, the test dataset consisted of $134 (= 140 - 3 \times 2)$ sounds for each ANR condition.

We evaluated the performance on the area under the receiver operating characteristic curve (AUC). In the same manner as [9], normal/anomaly was determined per each audio clip; the anomaly score of the whole frames of a test clip was calculated, and the test clip as determined as anomaly when the maximum value of the anomaly scores exceeded the threshold value. Table 1 shows the experimental results. Under both conditions, `SNIPER` outperformed `AE`, and the AUC scores of 3-shot situations were higher than that of 1-shot situations. Thus, overlooking anomalies significantly decreased and specific anomaly detectors succeeded in improving ADS performance. `SNIPER` also outperformed `COS-CE` and `MSE-CE` under both conditions. In registered categories, `SNIPER`'s AUC scores were higher than those of the conventional methods. These results indicate that the proposed method is effective for ADS.

Table 1. AUC results. Bold values mean highest scores under same condition, and underlined ones means highest scores under same ANR condition.

ANR: -15 dB

Method (#-shot, category)	door	pageturn	all
`AE` (baseline)	0.979	0.917	0.934
`COS-CE` (1-shot, door)	0.979	0.917	0.934
`MSE-CE` (1-shot, door)	**1.000**	**1.000**	0.964
`SNIPER` (1-shot, door)	**1.000**	**1.000**	**0.979**
`COS-CE` (3-shot, door)	**1.000**	0.882	0.900
`MSE-CE` (3-shot, door)	**1.000**	**1.000**	**0.987**
`SNIPER` (3-shot, door)	**1.000**	**1.000**	**0.987**
`COS-CE` (1-shot, pageturn)	0.979	0.917	0.933
`MSE-CE` (1-shot, pageturn)	0.979	0.972	0.913
`SNIPER` (1-shot, pageturn)	**1.000**	**1.000**	**0.988**
`COS-CE` (3-shot, pageturn)	0.976	0.920	0.945
`MSE-CE` (3-shot, pageturn)	**1.000**	**1.000**	0.953
`SNIPER` (3-shot, pageturn)	**1.000**	**1.000**	**0.995**

ANR: -20 dB

Method (#-shot, category)	door	pageturn	all
`AE` (baseline)	0.855	0.720	0.795
`COS-CE` (1-shot, door)	0.867	0.720	0.797
`MSE-CE` (1-shot, door)	0.962	**0.952**	0.881
`SNIPER` (1-shot, door)	**0.972**	0.917	**0.902**
`COS-CE` (3-shot, door)	0.882	0.675	0.759
`MSE-CE` (3-shot, door)	0.955	0.972	0.930
`SNIPER` (3-shot, door)	**0.976**	**0.975**	**0.931**
`COS-CE` (1-shot, pageturn)	0.855	0.720	0.794
`MSE-CE` (1-shot, pageturn)	0.892	0.675	0.780
`SNIPER` (1-shot, pageturn)	**0.969**	**0.969**	**0.928**
`COS-CE` (3-shot, pageturn)	0.851	0775	0.813
`MSE-CE` (3-shot, pageturn)	0.945	0.948	0.880
`SNIPER` (3-shot, pageturn)	**0.972**	**0.990**	**0.934**

The AUC scores of the others category, *i.e.,* "all", also increased with SNPER. The reason may be that $\mathcal{S}$ also includes dissimilarity of normal sounds. If $\boldsymbol{x}$ is not similar to $\boldsymbol{n}_i$, $\mathcal{S}$ returns a large value even though $\boldsymbol{x}$ is also not similar to $\boldsymbol{m}_k$. These results indicate that this does not deteriorate accuracy but rather improves it.

5. CONCLUSIONS

We proposed SNIPER; a few-shot learning method that works to minimize the FPR under the TPR when the TPR of the observed anomaly equals 1. To calculate the TPR, we used a VAE for simulating other samples of the observed anomaly. Experimental results indicate that SNIPER improved ADS performance over unsupervised-ADS and outperformed conventional methods. Thus, SNIPER is effective for few-shot learning for ADS to prevent overlooking.

6. REFERENCES

[1] C. Clavel, T. Ehrette, and G. Richard "Events Detection for an Audio-Based Surveillance System," *in Proc. of IEEE International Conference on Multimedia and Expo* (ICME), 2005.

[2] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and Gunshot Detection and Localization for Audio-Surveillance Systems," *in Proc. of IEEE International Conference on Advanced Video and Signal-based Surveillance* (AVSS), 2007.

[3] S. Ntalampiras, I. Potamitis, and N. Fakotakis "Probabilistic Novelty Detection for Acoustic Surveillance Under Real-World Conditions," *IEEE Transactions on Multimedia*, pp.713–719, 2011.

[4] 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*, pp.279–288, 2016.

[5] J. Kittler, I. Kaloskampis, C. Zor, Y. Xu, Y. Hicks, and W. Wang, "An intelligent signal processing mechanism for nuanced anomaly detection in action audio-visual data streams," *in Proc. of International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2018.

[6] A. Yamashita, T. Hara, and T. Kaneko, "Inspection of Visible and Invisible Features of Objects with Image and Sound Signal Processing," *in Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems,* (IROS), 2006.

[7] P. A. D. Arredondo, D. M. Sotelo, R. A. O. Rios, J. G. A. Cervantes, H. R. Gonzalez, and R. J. R. Troncoso, "Methodology for Fault Detection in Induction Motors via Sound and Vibration Signals," *Mechanical Systems and Signal Processing*, pp.568–589, 2017.

[8] Y. Koizumi, S. Saito, H. Uematsu, and N. Harada, "Optimizing Acoustic Feature Extractor for Anomalous Sound Detection Based on Neyman-Pearson Lemma," *in Proc. of European Signal Processing Conference* (EUSIPCO), 2017.

[9] 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*, 2018.

[10] V. J. Hodge and J. Austin, "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, pp. 85–126, 2004.

[11] A. Patcha and J. M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Journal Computer Networks*, pp.3448–3470, 2007.

[12] V. Chandola, A. Banerjee, and V. Kumar "Anomaly detection: A survey," *ACM Computing Surveys*, 2009.

[13] H. Lim, J. Park and Y. Han, "Rare Sound Event Detection Using 1D Convolutional Recurrent Neural Networks," *in Proc. of Detection and Classification of Acoustic Scenes and Events challenge* (DCASE), 2017.

[14] E. Cakir and T. Virtanen, "Convolutional Recurrent Neural Networks for Rare Sound Event Detection," *in Proc. of Detection and Classification of Acoustic Scenes and Events challenge* (DCASE), 2017.

[15] A. Mesaros, T. Heittola, A. Diment, B. Elizalde, A. Shah, E. Vincent, B. Raj, and T. Virtanen, "DCASE 2017 challenge setup: tasks, datasets and baseline system," *in Proc. of Detection and Classification of Acoustic Scenes and Events challenge* (DCASE), 2017.

[16] D. P. Kingma, and M. Welling, "Auto-Encoding Variational Bayes," *in Proc. of International Conference on Learning Representations* (ICLR), 2013.

[17] 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. of International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2015.

[18] T. Tagawa, Y. Tadokoro, and T. Yairi, "Structured Denoising Autoencoder for Fault Detection and Analysis," *Proceedings of Machine Learning Research*, pp.96–111, 2015.

[19] 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. of International Joint Conference on Neural Networks* (IJCNN), 2015.

[20] Y. Kawaguchi and T. Endo, "How can we detect anomalies from subsampled audio signals?," *in Proc. of IEEE International Workshop on Machine Learning for Signal Processing* (MLSP), 2017.

[21] J. An and S. Cho, "Variational Autoencoder based Anomaly Detection using Reconstruction Probability," *Technical Report. SNU Data Mining Center*, pp.1–18, 2015.

[22] Y. Kawachi, Y. Koizumi, and N. Harada, "Complementary Set Variational Autoencoder for Supervised Anomaly Detection," *in Proc. of International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2018.

[23] Y. Kawachi, Y. Koizumi, S. Murata and N. Harada, "A Two-Class Hyper-Spherical Autoencoder for Supervised Anomaly Detection," *in Proc. of International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2019.

[24] M. Yamaguchi, Y. Koizumi, and N. Harada, "AdaFlow: Domain-Adaptive Density Estimator with Application to Anomaly Detection and Unpaired Cross-Domain Transition," *in Proc. of International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2019.

[25] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," *in Proc. of Annual Conference on Neural Information Processing Systems* (NIPS), 2015.

[26] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillcrap, "Meta-learning with memory-augmented neural networks," *in Proc. of International Conference on Machine Learning* (ICML), 2016.

[27] S. Ravi, and H. Larochelle, "Optimization as a Model for Few-Shot Learning," *in Proc. of International Conference on Learning Representations*, (ICLR), 2017.

[28] `http://www.cs.tut.fi/sgn/arg/dcase2016/`

[29] `http://www.cs.tut.fi/sgn/arg/dcase2016/download`