MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Anomalous sound detection using attentive neural processes

Wichern, Gordon; Chakrabarty, Ankush; Wang, Zhong-Qiu; Le Roux, Jonathan

TR2021-129 October 21, 2021

Abstract

A typical approach for unsupervised anomaly detection of machine sounds learns an autoencoder model for reconstructing the spectrograms of normal sounds. During inference, fidelity of the reconstruction can be used to identify anomalous sounds different from normal sounds encountered during training. Recent improvements to the baseline autoencoder approach mask certain regions of the spectrogram at the input to the autoencoder, and then use the reconstruction error over masked regions as the anomaly score. We propose an alternative approach based on the attentive neural process, a recently proposed meta-learning technique for estimating distributions over signals. A benefit of our approach is that masked regions of the spectrogram do not need to be pre-specified at training time, and can determined based on signal properties or prior knowledge. Furthermore, we present an iterative approach that finds difficult-to-reconstruct spectrogram regions, and uses the reconstruction error over only those regions as the anomaly score. We demonstrate the effectiveness of our approach on experiments with the six machines of the DCASE 2020 Task 2 dataset, including in the case of zero-shot domain adaptation, where our approach outperforms baseline approaches in predicting anomalies for unseen machine instances.

IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) 2021

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

ANOMALOUS SOUND DETECTION USING ATTENTIVE NEURAL PROCESSES

Gordon Wichern, Ankush Chakrabarty, Zhong-Qiu Wang, and Jonathan Le Roux

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

{wichern, chakrabarty, zwang, leroux}@merl.com

ABSTRACT

A typical approach for unsupervised anomaly detection of machine sounds learns an autoencoder model for reconstructing the spectrograms of normal sounds. During inference, fidelity of the reconstruction can be used to identify anomalous sounds different from normal sounds encountered during training. Recent improvements to the baseline autoencoder approach mask certain regions of the spectrogram at the input to the autoencoder, and then use the reconstruction error over masked regions as the anomaly score. We propose an alternative approach based on the attentive neural process, a recently proposed meta-learning technique for estimating distributions over signals. A benefit of our approach is that masked regions of the spectrogram do not need to be pre-specified at training time, and can determined based on signal properties or prior knowledge. Furthermore, we present an iterative approach that finds difficult-to-reconstruct spectrogram regions, and uses the reconstruction error over only those regions as the anomaly score. We demonstrate the effectiveness of our approach on experiments with the six machines of the DCASE 2020 Task 2 dataset, including in the case of zero-shot domain adaptation, where our approach outperforms baseline approaches in predicting anomalies for unseen machine instances.

Index Terms— Anomaly detection, sound event detection, attentive neural process, autoencoder.

1. INTRODUCTION

Diagnosis and monitoring of machine operating performance is important for a wide variety of applications, and can often be performed by a skilled technician listening to the sounds produced by the machine. In order to automate this process, an algorithm that can process the sound signals produced by a machine and detect anomalies is desirable [1–3]. Unfortunately, collecting anomalous sounds for training supervised algorithms can be difficult, as it may require damaging an expensive piece of machinery, or impossible, as we may not know a priori the exact types of anomalous operating conditions that may occur. Thus, anomalies must be detected in an unsupervised manner, where learning algorithms are trained using only recordings obtained from observing the machine during normal operating conditions.

One class of methods for unsupervised sound anomaly detection augments the training data such that a surrogate supervised learning task becomes the training objective. Examples include outlier exposure [4,5], where sounds that are known to be very different from those of the observed machine are used as anomalous training examples; surrogate label prediction [6], where factors like different instances of the same machine, or the date when the recording was taken are used as labels; or self-supervised learning [7, 8], where a supervised classifier is trained to predict augmentations (e.g., time-stretching) applied to the input audio, and any sound predicted as being augmented at inference time is labeled an anomaly.

An alternative class of approaches, which we focus on in this paper, is based on the autoencoder (AE) [9], where a neural network learns to first compress and then reconstruct the normal training data, and any sounds that cannot be accurately reconstructed by the AE are considered anomalous. While various audio-specific network architectures have been used for sound anomaly detection [4, 9–11], two recent successful extensions of the autoencoder approach are particularly relevant to this paper. One is the interpolating deep neural network (IDNN) [12], where the model is trained to predict a given time frame of a spectrogram-like representation from only surrounding frames, leading to improvements over the basic autoencoder, especially for non-stationary sounds. The other is group masked autoencoder for density estimation (Group MADE) [13], which uses an autoregressive neural density estimator, where the audio anomaly score is computed using a likelihood of the true data with respect to Gaussian or mixture of Gaussian parameters estimated by the autoencoder. Both IDNN and Group MADE benefit from masking certain parts of the input and focusing the anomaly score on the reconstruction of only those masked regions. However, both IDNN and Group MADE require pre-specified masked regions at training time, and cannot adapt to properties of the input signal at inference time.

To increase the flexibility of masking-based AE approaches, we explore the neural process class of meta-learning models [14] for audio anomaly detection. The neural process estimates a stochastic process (i.e., model predictions also include estimates of uncertainty) for a set of target points from a context set of observed data points. By treating the context points as an un-ordered set, as opposed to pre-specifying an ordering as in autoregressive density estimators [13], and by encoding the coordinates (e.g., the bin indices of a spectrogram) along with the features (e.g, spectrogram magnitudes) at each point, the neural process can predict any masked target regions from arbitrary regions of provided context. While the vanilla neural process aggregates context information using a simple summation, recent extensions aggregate using attention [15] or convolution [16], providing more powerful models.

In this paper, we investigate the effectiveness of the attentive neural process (ANP) [15] for audio anomaly detection. Specifically, using log mel spectrograms as input, we explore various strategies for selecting subsets of time-frequency (TF) bins as context sets (i.e., network inputs) and target sets (i.e., predicted TF bins used for anomaly detection). Both the context and target sets can be chosen at inference time based on interpolation configurations known to work well such as that of IDNN, or using an iterative approach where multiple forward passes compute the reconstruction error over those TF regions that are most difficult to reconstruct. We demonstrate the effectiveness and flexibility of the proposed method on the six machines in the DCASE 2020 Task 2 dataset [17].

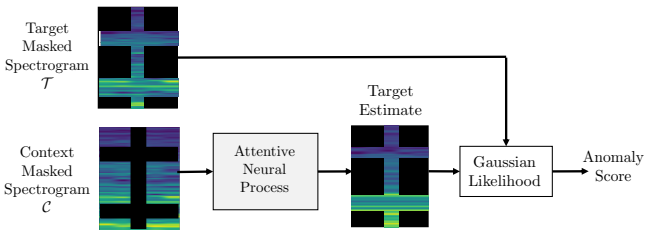

Figure 1: Example of partitioning a spectrogram into context and target sets for anomalous sound detection.

2. PROPOSED APPROACH

We aim to learn the parameters θ of an anomaly score function $\mathcal{A}_\theta : \mathbb{R}^{L\times F} \mapsto \mathbb{R}$ such that, given a sound signal produced by a machine and represented as a (log) magnitude spectrogram $Y \in \mathbb{R}^{L\times F}$, where L is the number of time frames and F the number of frequency bands, the score $\mathcal{A}_\theta(Y)$ is small for a normal sound and large for an anomalous one. An overview of our proposed approach is shown in Fig. 1, where we mask certain regions of the input spectrogram and use the quality of the reconstruction over those masked regions as the anomaly score.

2.1. Attentive neural process (ANP)

The ANP is an encoder/decoder model that can accommodate a flexible set of observed inputs (context set) and predicted outputs (target set) by encoding the coordinates of each element in the context set along with the observed value. It then learns to estimate conditionally independent Gaussian parameters for each element of the target set by attending to the context points of nearby coordinates as shown in Fig. 2. When the context set is small, far from the target points of interest, or different from the observed training data, we can expect the target set estimates to have high measures of uncertainty (i.e., variance), while low uncertainty estimates are expected for target points that align with the observed context.

In this work, we consider representations of audio signals such as mel spectrograms, where $\mathbf{x}_i = [\ell_i, f_i]^\top \in \mathbb{R}^2$ denotes the TF coordinates of bin i, and $\mathbf{y}_i = Y_{\ell_i,f_i} \in \mathbb{R}$ the magnitude at bin i. We partition the spectrogram into context set $\mathcal{C} = (\mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C}) = \{(\mathbf{x}_{c_j}, \mathbf{y}_{c_j})\}_{j=1}^{C}$, and target set $\mathcal{T} = (\mathbf{x}_\mathcal{T}, \mathbf{y}_\mathcal{T}) = \{(\mathbf{x}_{t_j}, \mathbf{y}_{t_j})\}_{j=1}^{T}$, where an example partition is shown in Fig. 1. The ANP then learns a model for the conditional distribution of the target values $\mathbf{y}_{t_j}$ given coordinates $\mathbf{x}_{t_j}$ and context set $\mathcal{C}$, assuming conditionally independent Gaussian distributions at each point in the target set:

$$p_\theta(\mathbf{y}_\mathcal{T}|\mathbf{x}_\mathcal{T}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C}) = \prod_{j=1}^{T} p_\theta(\mathbf{y}_{t_j}|\mathbf{x}_{t_j}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C}) \quad (1)$$

$$= \prod_{j=1}^{T} \mathcal{N}(\mathbf{y}_{t_j}; \mu_{t_j}, \sigma^2_{t_j}). \quad (2)$$

As illustrated in Fig. 2, we obtain the Gaussian parameters at each target point by first passing the concatenated coordinates and values of each context point through a self-attention encoder to obtain an encoding $\mathbf{r}_{c_j}$:

$$\mathbf{r}_{c_j} = \mathrm{Enc}_\theta([\mathbf{x}_{c_j}^\top, \mathbf{y}_{c_j}^\top]^\top). \quad (3)$$

We then compute the vector $\mathbf{r}_{t_j}$ summarizing the information in the context set most relevant to each bin t_j in the target set using

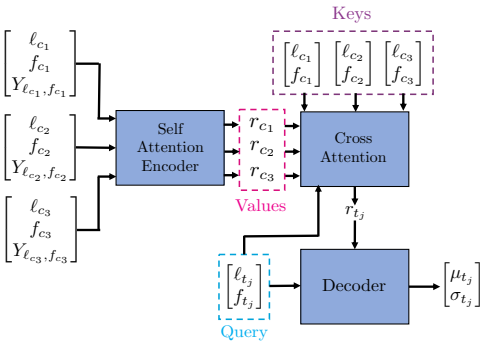

Figure 2: Block diagram of attentive neural process model.

cross-attention as

$$\mathbf{r}_{t_j} = \mathrm{Attention}_\theta(\mathbf{x}_{t_j}, \mathbf{x}_\mathcal{C}, \mathbf{r}_\mathcal{C}), \quad (4)$$

where $\mathrm{Attention}_\theta(Q, K, V)$ denotes multihead attention [18]. Finally, we obtain the Gaussian parameters μ_{t_j} and σ_{t_j} for each target point by passing the concatenation of the summarized context vector $\mathbf{r}_{t_j}$ with position vector $\mathbf{x}_{t_j}$ through a decoder network:

$$\mu_{t_j}, \sigma_{t_j} = \mathrm{Dec}_\theta([\mathbf{x}_{t_j}^\top, \mathbf{r}_{t_j}^\top]^\top). \quad (5)$$

The decoder has two output units [19], the first with a linear activation function for estimating μ_{t_j}, and the second with a regularized softplus activation to avoid the standard deviation collapsing to zero, i.e., $\sigma_{t_j} = 0.1 + 0.9 \cdot \mathrm{softplus(z)}$. All parameters are trained by maximizing the log-likelihood over all spectrograms in the training dataset $\mathcal{D}$:

$$\hat{\theta} = \underset{\theta}{\mathrm{argmax}}\, \mathbb{E}_\mathcal{D}[\log p_\theta(\mathbf{y}_\mathcal{T}|\mathbf{x}_\mathcal{T}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})]. \quad (6)$$

At inference time, the anomaly score for a given spectrogram is

$$\mathcal{A}_{\hat{\theta}}(Y) = -\log p_{\hat{\theta}}(\mathbf{y}_\mathcal{T}|\mathbf{x}_\mathcal{T}, \mathbf{x}_\mathcal{C}, \mathbf{y}_\mathcal{C})$$
$$\propto \sum_{t=1}^{T} \log(\sigma_{t_j}) + \frac{(\mathbf{y}_{t_j} - \mu_{t_j})^2}{2\sigma^2_{t_j}}. \quad (7)$$

If we ignore the variance by setting $\sigma_{t_j} \equiv 1$, then the anomaly score in (7) becomes the mean squared error (MSE) commonly used in AE-based anomaly detection.

2.2. Masking for context and target sets

A key factor in using the ANP to detect anomalies is partitioning the input spectrogram into context and target sets, both during training and inference. While the image examples in [14, 15] typically use a small number of randomly selected pixels as the context set and the entire image (including the context set) as the target set, we found this approach to be ineffective for anomalous sound detection. Instead, we focus on approaches that partition spectrograms based on entire time frames and/or frequency bands as shown in Fig. 1. In particular, we have found the following masks to perform well in our experiments.

Random RowCol (ANP-RRC): For training the ANP, we first randomly select one or two spectrogram columns (time frames) and up to two rows (frequency bands) as the target set, and use the remaining spectrogram bins as the context set, as illustrated in Fig. 1. At inference time, to obtain a relatively stable anomaly score, we average over three random context/target partitions.

Middle Frame (ANP-IDNN): While we train all ANPs using the ANP-RRC approach described above, the set-based formulation al-

lows flexibility in selecting different context and target sets at inference time. One approach, inspired by the success of interpolating a spectrogram frame from surrounding frames as in the IDNN [12], is to use the middle frame (frame $\frac{L+1}{2}$ for an L-frame spectrogram) as the target set, and the surrounding frames as the context set.

Likelihood bootstrapping (ANP-Boot): Instead of a fixed partition into context and target sets as with IDNN, we can run multiple forward passes with different context/target partitions with the goal of computing the anomaly score over only those frames and frequencies which are most difficult to reconstruct and therefore most likely to be anomalous. We use the following procedure:

1. Use $n_C\%$ of the spectrogram bins (uniformly downsampled) as the context set, and the rest as the target set. Based on preliminary experiments on development data, we use $n_C = 62.5\%$.
2. Identify the two rows (frequency bands) and the one column (time frame) with the lowest reconstruction likelihood.
3. Use the identified low-likelihood rows and columns as the target set, and the rest of the spectrogram as the context set.

3. EXPERIMENTAL RESULTS

3.1. Dataset

We evaluate our approach on the DCASE 2020 Task 2 dataset [17], which is composed of six machine types from the MIMII datset [20] (fan, pump, slider, valve), and the ToyADMOS dataset [21] (ToyCar, ToyConveyor). Anomalous sound samples are recorded by intentionally damaging the machines. For each machine type, there are multiple (six or seven) machine IDs corresponding to different instances of the same type of machine. Each audio sample is approximately 10 seconds long, with a sample rate of 16 kHz, and contains a small amount of real factory background noise.

The dataset for each machine type is categorized as follows: (1) DEV-train, (2) DEV-test, (3) EVAL-train, and (4) EVAL-test, where DEV and EVAL contain disjoint sets of machine IDs, train sets contain only normal examples (1000 per ID), and test sets contain a mixture of normal and anomalous samples (200-400 per ID). In the official DCASE 2020 Task 2 setup [17], systems are initially developed using DEV-train followed by re-training or fine-tuning on EVAL-train. We use a slightly modified setup, where we compare performance when machine IDs match between training and testing (e.g., DEV-train and DEV-test, EVAL-train and EVAL-test), with the case where IDs in the testing phase differ from those used in training, and fine-tuning on the new machines is not practical (e.g., DEV-train and EVAL-test). This situation can arise when it is required to detect anomalies in newly manufactured machines, or in high-volume manufacturing situations where fine-tuning for every instance of a machine may be impractical. As evaluation metrics, we use the area under the receiver operating characteristic curve (AUC) between the normal and anomalous sounds in the test set. The AUC indicates the probability that a randomly selected anomalous example will have a higher anomaly score than a randomly selected normal example. Additionally, we report the partial AUC (pAUC), which is the AUC computed under low false-alarm rate conditions, where $\mathrm{p} = 10\%$. We compute AUC and pAUC values separately for each machine ID and then average over machine IDs to obtain a representative score for each machine type.

3.2. Experimental Setup

We closely follow the setup in [17]. Our goal, however, is not to focus on the ensemble techniques necessary to obtain the best performance on this dataset, but rather evaluate the advantages and limitations of the ANP compared to existing AE and IDNN approaches. Specifically, we use log mel spectrograms as network input and output, computed with spectrogram frame length of 1024 samples, hop of 512, and 128 Mel bands. For all approaches, we use sliding windows of five frames with one frame hop as network input, and then average all windows to obtain the final anomaly score for each sample. As baselines, we consider the AE from [17], consisting of an encoder with four fully-connected layers of size 128, each with batch normalization and ReLU activations, a bottleneck layer with 8 units, and a decoder of similar configuration as the encoder, except with a linear output layer of appropriate size to reconstruct the input. We also modify the AE into the IDNN configuration, where we remove the middle frame from the input, and then train the network to reconstruct only that middle frame, which was shown to be the best configuration in [12]. Both the AE and IDNN use MSE as the training loss function and inference anomaly score.

For the ANP, we use the architecture for 2D inputs from [15], while setting all hidden layer dimensions to 128 to match the AE and IDNN baselines. Specifically, we set the dimension of the encodings $\mathbf{r}$ in (3) and (4) to 128. For the self-attention encoder, we first run each context point through three fully-connected layers of size 128 and then through two standard self-attention/transformer encoder layers [18]. We do not use dropout to limit the number of random confounding inputs to the model. The only source of randomness during training is in the selection of context and target sets. For the cross-attention block in Fig. 2, we run the query and key coordinate positions through two fully-connected layers of size 128 to obtain learned positional encodings prior to the cross-attention block, which also includes layer normalizations and a feedforward layer as is standard in transformer architectures. For both the cross-attention and self-attention blocks, we use multi-head attention with eight heads. The decoder contains three fully-connected layers with ReLU nonlinearities, except for the final output layer, which has the output units described in Section 2.1 for estimating μ and σ. For all systems, we used the ADAM optimizer with learning rate equal to 0.001, and adaptive gradient clipping [22]. Additionally, for the ANP we used the learning rate schedule from [18] with 4000 warmup steps. We train the ANP for each machine using the ANP-RRC approach from Section 2.2, and then compare different masking approaches at inference time.

3.3. Results

Table 1 compares the AUC and pAUC scores for all six machines on both the DEV and EVAL sets. We see that certain ANP configurations provide improved performance over the AE and IDNN baselines for most machines of the MIMII dataset, while not performing as well for the two machines from the ToyADMOS dataset under matched training conditions (either DEV-DEV or EVAL-EVAL). Also, similarly to what was observed in previous studies [12,13], we note that the largest gains are obtained for non-stationary machines such as valve and slider.

Comparison of ANP masking approaches: The different ANP masking approaches from Section 2.2 (ANP-IDNN, ANP-RRC, ANP-Boot) all performed best depending on the machine type as shown in the top (DEV-DEV) portion of Table 1. Surprisingly, ANP-RRC performed best only for pump, even though this was the strategy used for training the ANP. However, since the more structured inference strategies (ANP-IDNN and ANP-Boot) perform better, we only show EVAL set results for those approaches.

Table 1: AUC and pAUC for the six machines of the DCASE 2020 Task 2 dataset.

			ToyADMOS				MIMII							
			ToyCar		ToyConveyor		fan		pump		slider		valve	
System	Train	Test	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC	AUC	pAUC
AE	DEV	DEV	**80.9**	69.9	73.4	61.1	66.2	53.2	72.9	60.3	85.5	67.8	66.3	51.2
IDNN	DEV	DEV	80.3	**71.4**	**75.1**	**61.4**	67.7	53.1	**73.7**	61.3	86.6	67.1	84.5	64.0
ANP-IDNN-MSE	DEV	DEV	70.6	66.9	70.9	56.9	**71.3**	**55.2**	72.4	60.5	91.5	75.6	**93.7**	**76.7**
ANP-IDNN-Lik	DEV	DEV	70.7	65.9	70.2	56.8	69.3	54.6	69.4	60.9	93.9	81.4	91.5	71.7
ANP-RRC-MSE	DEV	DEV	72.5	67.3	67.0	54.5	69.2	54.4	72.8	**61.8**	90.7	74.2	86.9	70.7
ANP-RRC-Lik	DEV	DEV	72.5	66.6	66.8	55.0	67.1	54.2	69.0	59.7	92.9	79.8	88.0	72.7
ANP-Boot-Lik	DEV	DEV	72.9	68.1	67.1	54.2	64.8	53.0	65.5	59.0	**94.9**	**83.1**	85.2	72.0
AE	EVAL	EVAL	80.7	67.3	88.9	71.7	86.4	68.4	84.4	64.9	82.5	59.6	57.3	50.6
IDNN	EVAL	EVAL	**82.3**	71.1	**90.2**	**73.3**	88.2	70.9	83.5	65.0	83.2	59.9	77.2	57.9
ANP-IDNN-MSE	EVAL	EVAL	79.7	**75.6**	83.1	60.6	**89.9**	**72.4**	86.3	64.9	91.2	66.8	**78.7**	**58.6**
ANP-IDNN-Lik	EVAL	EVAL	80.5	75.4	83.9	62.1	89.3	69.8	**87.5**	**66.9**	93.7	73.0	72.1	55.0
ANP-Boot-Lik	EVAL	EVAL	77.9	70.5	68.4	53.6	85.4	65.6	82.8	63.8	**94.2**	**74.5**	72.0	56.8
AE	DEV	EVAL	59.3	53.9	45.3	50.3	49.3	50.8	**65.4**	58.5	71.3	55.8	37.1	49.1
IDNN	DEV	EVAL	63.5	57.2	50.7	50.5	51.5	50.7	64.1	**59.6**	72.1	55.8	41.2	**49.8**
ANP-IDNN-MSE	DEV	EVAL	**70.9**	**66.1**	59.6	**51.0**	**54.6**	52.2	63.0	58.0	77.9	60.1	41.5	48.8
ANP-IDNN-Lik	DEV	EVAL	70.7	65.8	60.1	50.9	54.4	**52.3**	60.1	56.5	82.4	63.3	35.3	48.7
ANP-Boot-Lik	DEV	EVAL	70.1	65.5	**60.1**	50.7	48.0	51.2	56.9	54.8	**85.4**	**68.4**	**43.5**	49.1

Table 2: Average mean squared error and standard deviation over all normal examples for different algorithms.

			ToyADMOS		MIMII			
System	Train	Test	ToyCar	ToyConveyor	fan	pump	slider	valve
AE	DEV	DEV	10.1 ± 0.4	10.3 ± 0.5	9.3 ± 1.2	10.3 ± 1.6	10.1 ± 1.8	9.7 ± 1.3
IDNN	DEV	DEV	10.3 ± 0.4	10.5 ± 0.5	9.7 ± 1.2	10.8 ± 1.7	10.4 ± 1.8	10.3 ± 1.4
ANP-IDNN	DEV	DEV	$\mathbf{9.5 \pm 0.3}$	$\mathbf{9.5 \pm 0.4}$	$\mathbf{8.9 \pm 0.6}$	$\mathbf{9.5 \pm 1.0}$	$\mathbf{9.4 \pm 1.0}$	$\mathbf{9.3 \pm 0.7}$
AE	DEV	EVAL	14.2 ± 1.9	17.6 ± 1.1	10.3 ± 1.3	11.0 ± 2.0	10.6 ± 1.7	$\mathbf{10.5 \pm 1.5}$
IDNN	DEV	EVAL	14.7 ± 1.7	16.7 ± 1.1	10.9 ± 1.4	11.5 ± 2.1	11.0 ± 1.7	12.8 ± 2.2
ANP-IDNN	DEV	EVAL	$\mathbf{11.6 \pm 0.6}$	$\mathbf{11.9 \pm 0.7}$	$\mathbf{9.4 \pm 0.6}$	$\mathbf{9.4 \pm 0.6}$	$\mathbf{9.8 \pm 1.1}$	11.3 ± 1.5

Impact of uncertainty estimation: We compare using the negative log likelihood from (7) as the anomaly score (denoted "-Lik" in Table 1) with a version where we ignore the variance estimates from the ANP and use MSE at inference time (denoted "-MSE" in Table 1). While using negative log-likelihood as the anomaly score is beneficial for the slider, many of the other machines exhibit similar or superior performance using MSE for the ANP anomaly score. This indicates that we may need to use more general distributions than a Gaussian, or investigate other methods that incorporate the uncertainty estimates more effectively.

Robustness to new machine IDs: In Table 1, we also consider the zero-shot domain adaptation situation, where systems trained on the DEV set are evaluated on unseen machine IDs from the EVAL set. In this case, the ANP approaches generalize much better than the AE or IDNN methods for ToyCar, ToyConveyor, and slider, slightly better for fan and valve, and slightly worse for pump.

To help analyze the increased robustness provided by the ANP, we compare the average reconstruction error for all normal spectrograms in Table 2. The ANP has lower reconstruction error for all machine types except valve in both domain matched (DEV-DEV) and mismatched (DEV-EVAL) conditions. Additionally, the relative difference in reconstruction error between matched and mismatched cases is generally much lower for the proposed ANP approach. We suspect the improved reconstruction performance can be attributed to the meta-learning aspect of the ANP. That is, unlike AE and IDNN which focus on minimizing reconstruction error for a given set of signals, the ANP methodology seeks to learn an underlying generative stochastic process, of which the training set is a realization. Therefore, for a given set of context points on unseen data, it can exploit statistical properties of the underlying distribution (rather than specific 'seen' instances as in AE/IDNN). A side effect of the ANP's strong reconstruction abilities suggests one of the main difficulties in unsupervised anomaly detection: robust reconstruction is important for the zero-shot domain adaptation scenario, but too much adaptability may be detrimental, as demonstrated by the relatively poor performance of the ANP approaches for the ToyADMOS DEV set in Table 1, where spectrograms produced by anomalous sounds are reconstructed well enough that they cannot be distinguished from normal sounds.

4. CONCLUSION

This work proposed a framework for anomalous sound detection based on the attentive neural process. Our approach extends previous approaches based on reconstructing or interpolating spectrograms of normal sounds by including encodings of TF bin locations in the network architecture, and treating the regions that we reconstruct as sets, allowing for a wide variety of different context/target configurations at inference time. Future extensions include exploring different variations of the neural process such as the convolutional [16] and sequential [23] variants, and estimating mixture model parameters for each TF bin as in [13, 24]. Finally, the inference-time flexibility and meta-learning aspects of the neural process should be useful in situations where a small number of anomalous sounds are observed as in [25, 26].

5. REFERENCES

[1] A. Ito, A. Aiba, M. Ito, and S. Makino, "Detection of abnormal sound using multi-stage GMM for surveillance microphone," in *Proc. IAS*, vol. 1, Aug. 2009, pp. 733–736.

[2] G. Valenzise, L. Gerosa, M. Tagliasacchi, F. Antonacci, and A. Sarti, "Scream and gunshot detection and localization for audio-surveillance systems," in *Proc. AVSS*, Sep. 2007, pp. 21–26.

[3] Y. Kawaguchi and T. Endo, "How can we detect anomalies from subsampled audio signals?" in *Proc. MLSP*, Sep. 2017, pp. 1–6.

[4] 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 Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 1, pp. 212–224, 2018.

[5] P. Primus, V. Haunschmid, P. Praher, and G. Widmer, "Anomalous sound detection as a simple binary classification problem with careful selection of proxy outlier examples," in *Proc. DCASE*, Nov. 2020, pp. 170–174.

[6] J. A. Lopez, H. Lu, P. Lopez-Meyer, L. Nachman, G. Stemmer, and J. Huang, "A speaker recognition approach to anomaly detection," in *Proc. DCASE*, Nov. 2020, pp. 96–99.

[7] T. Inoue, P. Vinayavekhin, S. Morikuni, S. Wang, T. H. Trong, D. Wood, M. Tatsubori, and R. Tachibana, "Detection of anomalous sounds for machine condition monitoring using classification confidence," in *Proc. DCASE*, Nov. 2020, pp. 66–70.

[8] R. Giri, S. V. Tenneti, F. Cheng, K. Helwani, U. Isik, and A. Krishnaswamy, "Self-supervised classification for detecting anomalous sounds," in *Proc. DCASE*, Nov. 2020, pp. 46–50.

[9] 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. ICASSP*, Apr. 2015.

[10] E. Cakır and T. Virtanen, "Convolutional recurrent neural networks for rare sound event detection," in *Proc. DCASE*, Nov. 2017.

[11] T. Hayashi, T. Komatsu, R. Kondo, T. Toda, and K. Takeda, "Anomalous sound event detection based on wavenet," in *Proc. EUSIPCO*, Sep. 2018, pp. 2494–2498.

[12] K. Suefusa, T. Nishida, H. Purohit, R. Tanabe, T. Endo, and Y. Kawaguchi, "Anomalous sound detection based on interpolation deep neural network," in *Proc. ICASSP*, May 2020, pp. 271–275.

[13] R. Giri, F. Cheng, K. Helwani, S. V. Tenneti, U. Isik, and A. Krishnaswamy, "Group masked autoencoder based density estimator for audio anomaly detection," in *Proc. DCASE*, Nov. 2020, pp. 51–55.

[14] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. Rezende, and S. A. Eslami, "Conditional neural processes," in *Proc. ICML*, Jul. 2018, pp. 1704–1713.

[15] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," in *Proc. ICLR*, May 2019.

[16] J. Gordon, W. P. Bruinsma, A. Y. Foong, J. Requeima, Y. Dubois, and R. E. Turner, "Convolutional conditional neural processes," in *Proc. ICLR*, Apr. 2020.

[17] 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 *Proc. DCASE*, Nov. 2020, pp. 81–85.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NIPS*, Dec. 2017, pp. 6000–6010.

[19] T. A. Le, H. Kim, M. Garnelo, D. Rosenbaum, J. Schwarz, and Y. W. Teh, "Empirical evaluation of neural process objectives," in *NeurIPS Workshop on Bayesian Deep Learning*, Dec. 2018.

[20] 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," in *Proc. DCASE*, Oct. 2019, pp. 209–213.

[21] Y. Koizumi, S. Saito, H. Uematsu, N. Harada, and K. Imoto, "ToyADMOS: A dataset of miniature-machine operating sounds for anomalous sound detection," in *Proc. WASPAA*, Oct. 2019, pp. 313–317.

[22] P. Seetharaman, G. Wichern, B. Pardo, and J. Le Roux, "AutoClip: Adaptive gradient clipping for source separation networks," in *Proc. MLSP*, Oct. 2020.

[23] J. Yoon, G. Singh, and S. Ahn, "Robustifying sequential neural processes," in *Proc. ICML*, Jul. 2020, pp. 10 861–10 870.

[24] W. J. Lee, K. Helwani, S. Tenneti, and A. Krishnaswamy, "Robust audio anomaly detection," in *RobustML Workshop at ICLR*, May 2021.

[25] Y. Koizumi, S. Murata, N. Harada, S. Saito, and H. Uematsu, "SNIPER: Few-shot learning for anomaly detection to minimize false-negative rate with ensured true-positive rate," in *Proc. ICASSP*, May 2019, pp. 915–919.

[26] Y. Koizumi, M. Yasuda, S. Murata, S. Saito, H. Uematsu, and N. Harada, "Spidernet: Attention network for one-shot anomaly detection in sounds," in *Proc. ICASSP*, May 2020, pp. 281–285.