Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-023-00792-2

RESEARCH

Audio-based anomaly detection on edge devices via self-supervision and spectral analysis

Fabrizio Lo Scudo[1] · **Ettore Ritacco**[2] · **Luciano Caroprese**[3] · **Giuseppe Manco**[4]

Received: 15 February 2023 / Revised: 21 April 2023 / Accepted: 23 April 2023

Abstract
In real-world applications, audio surveillance is often performed by large models that can detect many types of anomalies. However, typical approaches are based on centralized solutions characterized by significant issues related to privacy and data transport costs. In addition, the large size of these models prevented a shift to contexts with limited resources, such as edge devices computing. In this work we propose *conv-SPAD*, a method for *convolutional SPectral audio-based Anomaly Detection* that takes advantage of common tools for spectral analysis and a simple autoencoder to learn the underlying condition of normality of real scenarios. Using audio data collected from real scenarios and artificially corrupted with anomalous sound events, we test the ability of the proposed model *to learn normal conditions* and *detect anomalous events*. It shows performances in line with larger models, often outperforming them. Moreover, the model's small size makes it usable in contexts with limited resources, such as edge devices hardware.

Keywords Self-supervised learning · Audio-based anomaly detection · Edge-device computing · Pattern recognition

✉ Fabrizio Lo Scudo
fabrizio.loscudo@unical.it

Ettore Ritacco
ettore.ritacco@uniud.it

Luciano Caroprese
luciano.caroprese@unich.it

Giuseppe Manco
giuseppe.manco@icar.cnr.it

1 University of Calabria, Arcavacata, Italy

2 University of Udine, Udine, Italy

3 University of G. D'Annunzio of Chieti-Pescara, Pescara, Italy

4 ICAR Institute, National Research Council, Rende, Italy

Published online: 28 June 2023

1 Introduction

In the field of automatic surveillance, the initial focus on detecting anomalous events was mainly on video signals Haritaoglu et al. 2000. Although this type of signal is undoubtedly rich in information, its acquisition and processing may not satisfy some constraints, such as the availability of limited computational resources or the presence of privacy regulations that limit its use. In the attempt to overcome these limitations, Audio-based surveillance Clavel et al. 2019; Valenzise et al 2007; Ntalampiras et al. 2009, 2011; Foggia et al. 2016; Crocco et al. 2016 has emerged as alternative approach. The first advantage offered by the audio-based approach over the video-based one is that it requires less computational resources. Moreover, since ambient microphones can be easily deployed in the target environment, it also avoids the typical occlusion-related issues cameras suffer. Audio-based systems are unaffected by changes in lighting conditions that heavily affect the performance of video surveillance systems. Moreover, audio surveillance is perceived as less intrusive than camera systems, as it happens for the acoustic interface in smart environments Goetze et al. 2010 or in all situations in which privacy concerns are stressed Chen et al. 2005. Although audio surveillance systems usually do not include automatic speech detection and recognition, it does not mean that their use meets all privacy constraints. A complete discussion about the privacy issues of audio surveillance systems is out of the scope of this work, since it would involve an in-depth analysis of different legislation frameworks, such as the European one that includes the General Data Protection Regulation (GDPR) Kuner 2020, national programs for mass surveillance Bigo et al. 2020, and surveillance by intelligence services EU-Agency 2017.

In this work we assume that an audio surveillance system should have the following properties in order to avoid privacy issues:

- it does not require any human intervention;
- it does not involve any form of speech detection and recognition;
- it does not store any data since it processes and deletes the data;
- it is designed to accomplish the task locally with no communication to external resources.

Similar to Clavel et al. 2019, the audio surveillance system presented in this work aims to detect anomalous events occurring in a natural environment by exploiting acoustic information detected by a network of microphones. Recently, this line of research has made significant use of deep learning approaches Marchi et al. 2015a, b, 2017; Duman et al. 2019; Bayram et al. 2021; Suefusa et al. 2020; Kawaguchi et al. 2019.

As reported in Nunes (2021), complex neural network architectures, such as DenseNet-121 Papadimitriou et al. 2020, ResNet-50 Papadimitriou et al. 2020, WaveNet Rushe et al. 2019 or recurrent approaches Becker et al. 2020 are often adopted to deal with this task. Unfortunately, in contexts where only limited computational resources are available, their applicability is limited. A popular strategy to address this problem is based on the massive exploitation of cloud computing services Mayer et al. 2020, where the data collected by sensors are directly uploaded to external infrastructures with abundant computing and storage capacity. However, adopting such an approach has three main issues. First, it requires a stable connection to move the buffered sensor streams to data centers with appropriate strategies to handle high transmission delays. Second, the collected data may contain sensitive and private information about workplaces or individuals; thus, the data transfer to external infrastructures could raise significant concerns for data safety and privacy leakages Sánchez et al. 2016; Caire et al. 2016; Zhou et al. 2019. Finally, as the number of employed devices may proliferate, the Internet connection's bandwidth quickly becomes the system's bottleneck.

Thus, at scale, cloud-based computing could no longer be feasible or cost-effective: The high number of supervised environments would require transmitting an excessive amount of data. To alleviate the computational complexity of current deep learning approaches, we investigate the effectiveness of different strategies aimed at reducing the size of the models in order to move the computation from a centralized or cloud infrastructure to the deployed edge devices. Even though the computational power of edge devices has constantly been increasing, it is still not enough to support a wide range of deep learning architectures.

For this reason, we study and propose specific approaches to analyze the environmental sounds directly on the devices, trying to balance the detection capability of unexpected events and the computational cost in different application scenarios.

In practice, we exploit the synergy between the *Edge Computing*, the *Internet of Things* and the *Artificial Intelligence* Greengard 2010; Lee et al. 2018. In our framework, the *edge devices* are physical devices equipped with sensing, computing, and communication capabilities, able to buffer sensor observations, discover underlying patterns and unexpected anomalies using no external resource and share the detected data with other systems.

In order to match the requirements imposed by the limited computational resources, in this work, we investigate state-of-the-art solutions and perform an iterative analysis to define a lightweight neural network for unsupervised audio-based anomaly detection. Since we aim for audio-based anomaly detection suitable to applications in various real-life scenarios, we can only perform a partial feature domain search in the context of time and frequency domains. We then limit our analysis to only the input audio signal's temporal dimension while focusing on a deeper evaluation of audio features in future work. We also do not search for the best-performing neural architecture. Instead, we decide to use common building blocks and standard training strategies. Further improvements to the proposed architecture will be provided in future works. Thus, the aspects we will analyze are:

- *Input manipulation*. We will show how good sizing of the incoming audio signals can significantly contribute to the detection capability of the model and the reduction of its computational cost.
- *Transfer learning*. We will discuss how using pre-trained networks to embed in audio-based anomaly detection solutions does not provide consistent performance improvements in the face of more computational requirements.

We prove our claims by conducting an in-depth analysis on several real benchmark datasets.

The rest of the paper is structured as follows. Section 2 describes the state of the art in the literature about the audio-based anomaly detection problem. In Section 3 we introduce our research questions, and in Section 4 we propose a lightweight model for audio-based anomaly detection. In Section 5 we present our experimental results. Finally, Section 6 summarizes our findings and describes future work.

2 Related work

Anomaly detection Hodge and Austin 2004; Patcha et al. 2007; Chandola et al. 2009 is the ability to detect unexpected events that significantly deviate from those assumed to describe normal conditions. In unsupervised settings, anomaly detectors have to operate where no semantic label about the system's status is available. Therefore, they aim at recognizing *patterns of normality* to isolate anomalies.

Recently, approaches based on deep learning techniques have rapidly become popular, with the encoder-decoder architecture as the dominant one. For instance, Marchi et al. 2015a, b, 2017 focus on acoustic novelty detection within various environments through a sequence-to-sequence approach to reconstruct auditory spectral features. In contrast, Duman et al. 2019; Bayram et al. 2021 prefer to focus deeply on the specific task of identifying acoustic anomalies in industrial plants by exploiting both sequential or convolutional Autoencoders. In Koizumi et al. (2019a) authors, exploiting another Autoencoder architecture, recast the unsupervised anomaly detection problem as a statistical test. Finding an anomaly corresponds to rejecting the null hypothesis "a sequence belongs to a normality pattern". The hypothesis is rejected according to an objective function based on the Neyman-Pearson lemma Neyman et al. 2018 that aims at increasing the anomaly prediction's actual positive rate while keeping the false positive low rate. As a follow-up, in Koizumi et al. (2019b), the same authors investigate the issue of dealing with overlooking anomalies without retraining the whole system. They propose a training method for a cascaded specific anomaly detector using few-shot samples.

Different approaches to anomaly detection are presented in Suefusa et al. (2020); Kawaguchi et al. (2019). The former feeds a completion (Variational) Autoencoder with multiple frames of the input spectrogram whose centre frame is removed. The technique's objective is to reconstruct the missing frame and compare the reconstruction with the actual frame: if the difference is above a threshold, an anomaly is identified. The latter investigates how detection performance degrades due to reverberation and factory background noise affecting machine signals. They propose a method based on a front-end ensemble of algorithms for de-reverberation and de-noising to improve detection performance.

Most of the state-of-the-art approaches are based on the extraction of the spectrogram of the original signal. However, in some recent work, the literature is experimenting with a new trend by operating directly on the raw audio waveform. Following this idea, Hayashi et al. 2018; Rushe et al. 2019 define encoder-decoder architectures exploiting causal dilated convolutions and, given a sequence of audio samples, they try to predict the following sample. In Rushe et al. (2019), authors investigate the use of auto-regressive deep learning architectures for anomaly detection: they use a WaveNet architecture Oord et al. 2016, which was previously proposed as an auto-regressive approach to speech synthesis to predict the following typical sample in a sequence. They thus try to learn a conditional distribution for standard sequences under the assumption that anomalous sequences should not follow the same distribution. Being an auto-regressive approach, the proposed solution is intrinsically slower than other solutions, although, as reported in the paper, there exist some techniques to speed up the process van den Oord 2018; Paine et al. 2016.

The approaches discussed so far were proved effective to detect anomalies in specific environments. However, this paper aims to address the more general problem of ambient surveillance, which is characterized by quite varied audio signatures compared to detecting anomalies in controlled scenarios. As a strict constraint, we notice that surveillance requires a limited consumption of time, memory, energy and resources: this leads to the need to focus on very light techniques that can run fast on limited hardware. The solution currently closest to our problem is Rushe et al. (2019), which is incapable of working under limited hardware constraints due to the underlying network size. Although designed to cope with a different goal (speech synthesis), the WaveNet approach proposed there is the only one, to the best of our knowledge, is comparable with our proposal. We thus compare our design choices with WaveNet in the following. In order to make a fair comparison, we do not employ sophisticated layers to process the spectrogram and limit ourselves to only standard fully-connected and convolutional layers. Moreover, in Section 5.2, we reverse the claim made in Rushe et al.

(2019) and show that thanks to the spectral analysis, even a simple convolutional Autoencoder can obtain a performance gain over the auto-regressive approach.

3 Design choices

The analysis we are going to make takes advantage of many state-of-the-art results. The goal is to navigate the literature about the acoustic anomaly detection problem to build a lightweight architecture that can be equipped on edge devices with limited resources. In the following, we address one issue at a time, providing our qualitative (in this section) and quantitative (in Section 5) answers and motivations.

3.1 Q1. Which transformation of the input audio better fits the anomaly detection problem?

As mentioned in Section 2, many research proposals used to work with small audio segments. We speculate that such a choice is mainly driven by the application domain (most of those works are focused on industrial machine failures) or by architectural choices (recurrent or auto-regressive models). Moreover, using short audio segments involves complex detection models that cannot be exploited in real scenarios. On the other hand, we hypothesize that, for ambient surveillance settings, longer input audio samples are necessary. In many cases, anomalous sounds last for seconds, so larger window frames could provide richer information for detecting unexpected patterns: short samples risk splitting a target pattern into little pieces unable to trigger any relevant alert.

Moreover, we assume that standard audio signal manipulation procedures, such as frequency domain transformations and windowing operations, are feasible on edge devices. Therefore, throughout the spectral analysis, we will shift the form of the input from the time domain to a time-frequency representation in which valuable information about frequency patterns is easier to detect. This kind of representation is widely used in different audio tasks ranging from audio classification to automatic speech recognition (Hinton et al., 2012; Oruh et al., 2021). Its success is mainly tied to the fact that it simplifies the training process thanks to the presence of evident patterns related to frequency components. Thus we assume a spectrogram as input for our models.

3.2 Q2. Which network topology should we use?

As we said, the most popular solutions addressing our problem are based on encoder-decoder neural architectures (mainly Autoencoders Baldi 2012 and Sequence to Sequence models Sutskever et al. 2020). Those can mainly exploit three network topologies: recurrent, feed-forward, and convolutional Li et al. 2022.

Recurrent networks are the most natural choice to learn patterns from timed signals since they build up a *history* of the data by ingesting the sampled values according to chronological order. However, they are extremely time-consuming, since, in the most effective implementation (e.g. LSTM Hochreiter et al. 1997, GRU Cho et al. 2014), each node unit is a complex sub-network.

On the other hand, feed-forward networks are typically fast since they are based on the bag-of-word assumption. They do not consider the time relationship between the samples of the signals; in other words, these networks can have only a partial view of the information

a signal contains. Moreover, feed-forward networks are fully connected graphs implying a big number of parameters to learn and store, making this technology not feasible in limited hardware settings.

Convolutional networks, instead, represent a good compromise between the former topologies. Moreover, in their causal variant Brazil 2013, these networks can model temporal sequences limiting the number of parameters (w.r.t. feed-forward networks) and the number of iterations (w.r.t. recurrent networks). For these reasons, we decided to adopt the encoder-decoder technology based on convolutional networks to develop a lean and effective anomaly detection model.

3.3 Q3. Is transferred knowledge useful?

A typical approach to improve the detection quality, or in general the data fitting, is to exploit transferred knowledge from systems dealing with data belonging to a similar domain. Lately, this knowledge is often stored in pre-trained neural networks that are typically used as embedding modules to map data into suitable latent spaces. The usefulness of this approach arises when the latent space is built for highlighting specific patterns related to the target task. However we make three considerations:

1. Pre-trained networks are often heavy data structures composed by many complex layers with several connections and parameters;
2. Sometime, they need to be involved in the learning phase of the global network: in these cases, their parameters are considered as an initialization strategy for an optimal selection of the solution space, but they may represent another issue for the efficiency of the detection system;
3. They may store much more information than needed: in many surveillance scenarios, anomalies are disruptive events that strongly deviate from normality. Hence, exploiting a heavy pre-trained neural network could be an overpowered answer to the detection problem.

For these reasons, we decided to verify the impact of using a pre-trained network and design a simple model that is fed with the output of a pretrained network. We thus compare a transfer-learning approach with a simpler convolutional autoencoder trained from scratch. As we will show in Section 5, introducing a feature extractor does not consistently improve the performance when requiring a consistent amount of computational resources.

4 A lightweight model for audio-based anomaly detection

In this section, we describe three different models for audio-based anomaly detection that we want to compare in search of a suitable solution for edge device computing. We define the three models, namely *fully-SPAD+CNN14*, *fully-SPAD* and *conv-SPAD* (where the acronym *SPAD* means *SPectral audio-based Anomaly Detection*), showing that the latter ensures good performance while being much lighter than the others. However, first, we discuss how we transform the input audio signal for our analysis.

Figure 1 shows how the input audio signal is pre-processed before becoming the input for our models. The first step is to apply a sliding window procedure to segment the raw audio signal into suitable time frames of fixed size. Then, from each time frame, a Short Time Fourier Transform (STFT) module (Oruh et al., 2021) extracts a spectrogram that is represented by a matrix in $\mathbb{R}^{F \times T}$, where F is the number of frequency bins and T the number of time ticks.

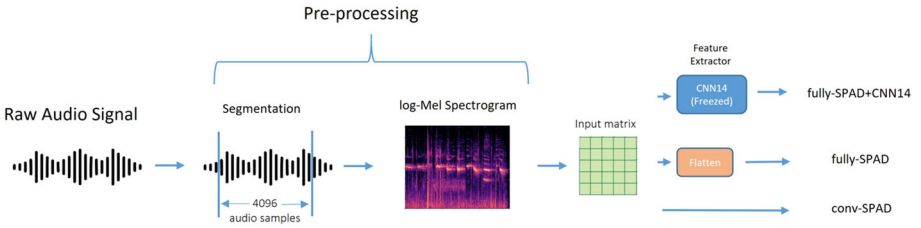

Fig. 1 The pre-processing phase segments the raw audio signal and extracts the log-Mel spectrogram. The spectrogram will be either i) feed into the feature extractor in order to obtain the actual input for *fully-SPAD+CNN14* or ii) flatted to obtain the actual input for *fully-SPAD*, or iii) used unchanged by *conv-SPAD*

Each cell contains the amount of energy related to a specific frequency at a certain tick within the spectrogram. After, a further transformation of the spectrograms is performed (without changes to the size of the matrix). It consists in exploiting a bank of triangular filters to produce *log-Mel Spectrograms* (Imai, 1983; Hinton et al., 2012). These filters aggregate the energy of consecutive frequencies mimicking the human sound perception and making it easier for models to solve the typical audio-based tasks a human can be interested in. Log-Mel spectrograms are the processed input to feed and train our audio anomaly detectors. We can now introduce the details of our models.

The three models we present are all based on the Autoencoder architecture. An autoencoder is a neural network trained to attempt to replicate its input as output. This architecture aims to learn some low-dimensional feature representations from which the input data instance can be reconstructed. Although this approach is often used for data compression or dimensionality reduction (Hinton et al., 2006; Theis et al., 2017), it can be employed for anomaly detection by learning latent representations enforced to capture important regularities across the training data.

In our case, a trained autoencoder mainly learns to replicate spectrograms representing frequency patterns in the monitored environment under typical conditions. We thus train our models with input audio signals without anomalous events. As a result of this training strategy, anomalies, assumed as rare events, will be challenging to replicate and, as a consequence, will produce a high reconstruction error.

Formally, an autoencoder uses two networks $\mathbf{E}$ and $\mathbf{D}$ as encoder and decoder respectively. The feature input $\mathbf{x}$ is converted into the latent code $\mathbf{z} \in \mathbb{R}^m$ by $\mathbf{E}$. This code is later used by $\mathbf{D}$ to reconstruct the original input:

$$\begin{aligned} \mathbf{z} &= \mathbf{E}\left(\mathbf{x}; \theta_{\mathbf{E}}\right) \\ \widehat{\mathbf{x}} &= \mathbf{D}\left(\mathbf{z}; \theta_{\mathbf{D}}\right) \end{aligned} \tag{1}$$

where $\theta_{\mathbf{E}}$ and $\theta_{\mathbf{D}}$ are, respectively, the encoder an decoder parameters learnt during the training. The autoencoder's loss function aims at minimizing the average reconstruction error upon the normal input which is expressed in term of mean squared error (MSE):

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2 \tag{2}$$

where N is the number of the input log-Mel spectrograms.

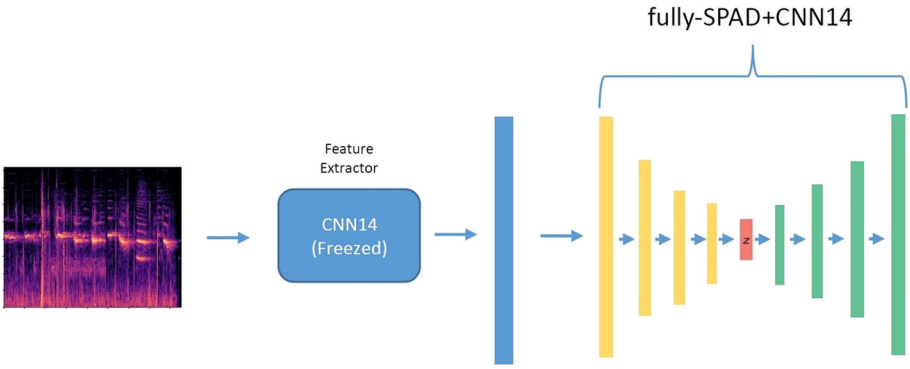

Fig. 2 In fully-SPAD+CNN14 the autoencoder has four fully-connected layers for encoding and four for decoding. The objective is to reconstruct the feature vector produced by the feature extractor

4.1 Fully-SPAD+CNN14

The first model we propose, Fully-SPAD+CNN14 (Fig. 2), is composed of two sub-networks: a frozen pre-trained feature extractor network and a shallow autoencoder. As feature extractor we choose a recent architecture presented in Kong et al. (2020). In this work, the authors try to provide a general model for audio pattern recognition by training different CNNs for the audio classification task over the large audio dataset AudioSet Gemmeke et al. 2017. Among the architectures proposed in that work, we selected the model called CNN-14 as our feature extractor. This model has five convolutional blocks based on 3×3 filters, batch-normalization and Relu. From this architecture, however, we removed the last layer, specific for the classification task over AudioSet, and used the network's body output as audio features, which are expressed as vectors. The encoder is a sequence of simple fully connected layers with a ReLu function. The decoder mirrors the encoding part, and the parameters of both networks are learnt by minimizing (2) loss function over the output of the feature extractor.

4.2 Fully-SPAD

Fully-SPAD (Fig. 3) is a variant of the previous model. We removed the feature extractor and introduced a flatten module that transforms the log-Mel spectrograms into vectors by concatenating the rows. Then the model presents additional fully connected layers to the encoder and the decoder. In doing so, we pay an extra cost in terms of an increased number of parameters of the model, but we tried to simulate the contribution of the extractor through the added layers. In this case, the encoder and decoder are trained by minimizing (2) loss function over the flattened log-Mel spectrograms.

4.3 Conv-SPAD

The last model is Conv-SPAD (Fig. 4) which replaces the dense connections of Fully-SPAD with a combination of convolutional layers. All the convolution operators are bi-dimensional and have 3×3 kernels with stride $(1, 2)$ and padding 1. In the figure as example, we report, for each convolution layer, the dimension of the resultant blocks of feature maps for the input size 16834 whose size is $\langle 1, 52, 64 \rangle$ with a frequency representation divided into 52 different

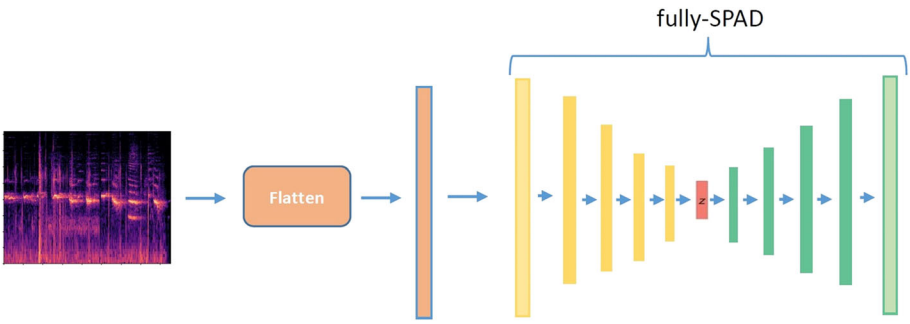

Fig. 3 In fully-SPAD the autoencoder has a total of ten fully-connected layers to accommodate the larger input. The objective is to reconstruct the flattened log-Mel spectrogram

time step and 64 frequency bins. In Fig. 4, we use the notation $ch@h \times w$ where ch represents the number of channels, h the temporal dimension and w the frequency dimension of the feature map matrix. At the core of the Autoencoder there are two symmetric fully-connected layers: the first encodes the output of the convolutional part into a latent code z, whose size is 64, and the second decodes that code into the input for the transposed convolutions. The objective is to reconstruct the original log-Mel spectrogram matrix. In this model, the number of parameters grows much slower than the fully-SPAD, but, as we will discuss in Section 5, this model capacity reduction slightly affects the model's overall performance. The loss function of Conv-SPAD aims at minimizing (2) directly over the log-Mel spectrograms.

5 Experiments

5.1 Training and testing settings

In the context of Research Question ***Q1***, we conduct experiments with a variable input duration to highlight the effect of the temporal horizon on the anomaly detection system's performance. In the subsequent experiments, we evaluate four different time scales: 0.1 seconds, 0.4 seconds[1], 1 second and 2 seconds. For an audio signal sampled at 44100 Hz these choices correspond to 4096, 16384, 44100 and 88200 audio samples, respectively. There is no overlapping between samples.

We perform experiments on two datasets: *TUT Rare Sound Events 2017*[2] and *TAU Urban Acoustic Scenes 2020* datasets [3]. In each experiment, the data used for training do not contain any anomalous events. Thus everything which is fed into the models is assumed to be part of the normality condition. As a hyper-parameter setting, we use the Adam optimizer with a learning rate of 10^{-4}, a batch size of 128 and a number of epochs equals to 100 for the fully-SPAD+CNN14 and conv-SPAD while 250 for the fully-SPAD. For the testing phase, we consider an anomaly to be any instance belonging to the rare event sounds available in the TUT dataset. Whereas with this dataset, we can count on the provided mixtures of

[1] The minimum length the original CNN-14 model accepts at inference time is 16384 audio samples. We empirically found this value.

[2] https://zenodo.org/record/401395 and https://zenodo.org/record/1160455

[3] https://zenodo.org/record/3670167

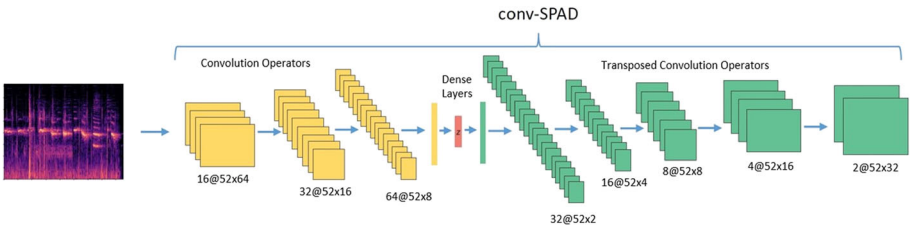

Fig. 4 In conv-SPAD the autoencoder is asymmetric with three layers of bi-dimensional convolutions for the encoding and five transposed bi-dimensional convolutions for the decoding phase. Here, we use the notation $ch@h \times w$ where ch represents the number of channels, h the temporal dimension and w the frequency dimension of the feature map matrix

background sounds and rare events, for the TAU dataset, we had to manually inject anomalous events from TUT into these new background sounds. We randomly perform the mix with the subsequent procedure inspired by the mixing procedure used for the TUT Dataset. We inject a randomly chosen event with a prefixed probability for each background sound. We also randomly select the offsets within the event and for the inject point. This way, we try to introduce a high degree of randomness during the mixing procedure. The performance of each model is then evaluated using the area under either the ROC curve and the precision-recall curve, while the model sizes (in terms of number of parameters) is reported in Table 1. The code for the experiments will be released at the link: https://github.com/fablos/spad.

5.2 TUT rare sound events 2017 dataset

The TUT Rare Sound Events 2017 dataset Mesaros et al. 2017 provides recordings of 15 everyday acoustic scenes which serve as background and separately provides sounds about isolated rare events belonging to three different classes: *baby-cry*, *glass-break* and *gun-shot*. The overall dataset is split into a training set and an evaluation set. The former contains 9 hours of the background sounds and 100 events from each class, while the latter contains 12 hours of background and 500 events for each class. In the training phase, we ignore the rare classes and feed the detection models with the background noise. The test set is instead built by randomly mixing background and events, following an event occurrence probability and different event-to-background ratios, defined as the ratio between the root mean square energy of the rare event and the background audio. Upon this data, we apply the

Table 1 The number of parameters for the AEs evaluated in the analysis

	Num. of parameters (Size in bytes)			
Input length	4096	16384	44100	88200
fully-SPAD+CNN14	–	5.5M (21MB)	5.5M (21MB)	5.5M (21MB)
fully-SPAD	8.9M (34MB)	19.2M (73MB)	41.7M (159MB)	78M (297MB)
conv-SPAD	155k (607KB)	477k (1MB)	1.2M (4MB)	2.3M (8MB)

With fully-SPAD+CNN14 we have to consider to add the CNN14 cost which is 80.7M parameters (equivalent to 312MB)

Table 2 AUC and AUC PR results for Wavenet and SPAD on 4096 samples input

Model	WaveNet		fully-SPAD		conv-SPAD	
place	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR
beach	0.72	0.29	**0.81**	**0.46**	**0.81**	**0.46**
bus	0.83	0.48	**0.87**	**0.57**	0.85	0.56
cafe restaurant	**0.76**	**0.24**	0.72	0.13	0.73	**0.24**
car	0.82	0.61	**0.90**	**0.66**	0.89	0.65
city center	0.82	0.32	**0.85**	**0.50**	0.75	0.21
forest path	0.72	0.09	**0.77**	**0.19**	0.74	0.18
grocery store	0.77	0.24	**0.79**	0.33	**0.79**	**0.44**
home	0.69	0.07	0.70	0.10	**0.71**	**0.16**
library	0.67	**0.18**	0.75	0.10	**0.76**	0.13
metro station	**0.79**	0.14	**0.79**	**0.38**	0.70	0.06
office	0.78	0.21	0.79	0.41	**0.80**	**0.44**
park	0.80	**0.46**	**0.83**	0.45	0.82	0.44
residential area	**0.78**	0.22	**0.78**	**0.31**	0.76	0.30
train	**0.84**	0.46	0.83	**0.51**	0.83	0.50
tram	**0.87**	0.56	**0.87**	**0.64**	0.86	0.62
Overall Score	0.78	0.3	**0.8**	**0.38**	0.79	0.36

The best results are marked in bold

pre-processing procedure described in Section 4 and compare fully-SPAD+CNN14, fully-SPAD and conv-SPAD with the WaveNet model presented in Rushe et al. (2019).

The first experiment in our analysis is on an input length of 4096 audio samples, equivalent to about 100ms. In this case, we cannot use fully-SPAD+CNN14 since its minimum admissible input length is 16384. In Table 2 we report the results, in terms of Area Under ROC Curve (AUC) and Area Under Precision-Recall Curve (AUC PR), obtained on the different background environments in TUT dataset.

The results show that both the models fully-SPAD and conv-SPAD have comparable results with WaveNet. This latter is able to reach better performance in only two cases: *cafe restaurant* and *train*. In all the other cases its results are either matched (3 cases) or improved (10) by the model fully-SPAD. At the same time, the model conv-SPAD obtains comparable results with both the other two models across the different environments; moreover it is able to improve the WaveNet's overall score both on the AUC and the AUC PR. It is important to note here that, while the WaveNet and fully-SPAD have comparable sizes[4], the conv-SPAD is much smaller than those models (see Table 1). Nevertheless, its performances are still comparable with the other competitors, except for only two environments: *city center* and *metro station*. These results support our claim about the advantage of the frequency analysis as discussed in Research Question ***Q1***.

We continue our analysis by increasing the input to 16384 audio samples, the minimum input length accepted by CNN14. We thus keep unchanged the parameters of the feature extractor and only trained the Autoencoder module of fully-SPAD+CNN14, presented in Section 4. The results of these experiment are shown in Table 3.

[4] The WaveNet has about 12M parameters, while the fully-SPAD has around 9M parameters.

Table 3 AUC and AUC PR results for SPAD and Wavenet 16384 Samples Input

Model	WaveNet		fully-SPAD+CNN14		fully-SPAD		conv-SPAD	
place	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR
beach	0.74	0.34	0.82	0.28	0.84	0.50	**0.85**	**0.54**
bus	0.85	0.55	0.87	0.46	**0.89**	**0.67**	0.87	0.63
cafe restaurant	**0.76**	0.30	0.73	0.21	**0.76**	**0.34**	0.74	0.15
car	0.84	0.68	0.88	0.49	**0.91**	**0.70**	.90	0.69
city center	0.83	0.34	**0.87**	**0.56**	0.86	0.46	**0.87**	0.55
forest path	0.73	0.11	**0.79**	**0.20**	0.76	0.14	0.75	0.19
grocery store	0.77	0.26	0.81	0.27	0.83	0.33	**0.84**	**0.50**
home	0.69	0.09	**0.72**	**0.15**	**0.72**	0.08	**0.72**	0.10
library	0.67	0.18	**0.75**	**0.24**	0.73	0.10	0.74	0.14
metro station	0.79	0.16	0.80	0.35	0.82	0.40	**0.83**	**0.42**
office	0.80	0.23	0.81	**0.44**	**0.83**	0.42	0.82	0.30
park	0.80	0.47	**0.85**	0.39	0.84	**0.53**	0.72	0.32
residential area	0.77	0.12	0.79	0.26	**0.82**	0.36	0.80	**0.39**
train	**0.86**	0.51	0.83	0.34	**0.86**	0.51	0.85	**0.53**
tram	**0.89**	0.65	0.83	0.47	0.87	0.62	0.87	**0.66**
Overall Score	0.79	0.33	0.81	0.34	**0.82**	**0.41**	0.81	**0.41**

The best results are marked in bold

With this new input size, WaveNet slightly improved its performance in a few environmental scenes as well as SPAD models (namely fully-SPAD+CNN14, fully-SPAD and conv-SPAD), supporting our claim about segment lengths concerning Research Question ***Q1***. WaveNet wins only on one scene, *tram*. At the same time, all its other results are matched (*cafe* and *train*) or improved by SPAD models (12 out of 15 scenes). The results from Table 3 also show similar performance between the two models fully-SPAD and conv-SPAD and the model fully-SPAD+CNN14. We expected that a feature extractor, such as CNN14, would provide representations more beneficial for the task. However, the results show only few slight improvements (except for *library* AUC PR) while in the majority of the scenarios fully-SPAD+CNN14 is equaled by the competitors or outperformed. This suggests that exploiting the transferred knowledge may be redundant (see Research Question ***Q3***). The following experiments further confirm this last point.

We now proceed to test the SPAD models using an even longer input in order to discover how such a richer input may help to reach even better results, providing an empirical answer to Research Question ***Q1***. We argue that long inputs do carrier more informative content that should allow the models to be more exposed to the effect of anomalies. Since we are injecting anomalous events in our tests, whose average duration is about two seconds, we hypothesize that a lower variability characterizes small inputs than long ones. Thus the models would still be able to reconstruct the input even in the presence of anomalies. On the other hand, long inputs, having a signature built over a longer temporal horizon, should be characterized by a structure more conditioned by anomalous patterns, making the reconstruction process harder. We thus continue our analysis with input whose duration is 1s (44,100 samples) and 2s (88,200 samples).

In Table 4 we report the results for only the SPAD models (fully-SPAD+CNN14, fully-SPAD and conv-SPAD) since we experienced memory-related issues[5] with the WaveNet model which prevent us to properly train this model.

The results of these experiments show how the larger models perform better in solving the task. That is probably related to their greater capacity to model the data distribution, which is now more complex than in the previous experiments. This aspect is confirmed by a consistent drop in performances of the conv-SPAD with the 2s input over some scenarios. We can see the same behaviour also with the 1s-long segments but with slighter effects. Overall, however, all the remaining models seem to benefit from the long inputs showing smooth performance improvements.

Unfortunately, improvements in quality come at the cost of heavier models. By looking at the size of the models in Table 1, we can see how all the models but conv-SPAD may exceed the limits imposed by extreme edge computing. We thus consider the conv-SPAD with 1s input as a good compromise between performance and model complexity, answering both Research Questions ***Q2*** and ***Q3***.

5.3 TAU Urban Acoustic Scenes 2020 dataset

The TAU Urban Acoustic Scenes 2020 dataset is an updated version of the TAU Urban Acoustic Scenes 2019 dataset (Mesaros et al., 2018) which was used for the task of acoustic scene classification in the DCASE 2019 [6].

In this dataset, ten acoustic scene classes are recorded in twelve large cities. In our experiments, however, we found some issues with the files related to two cities, Amsterdam and Madrid, so we decided to remove them.

Each acoustic scene has 1440 10-second segments, equivalent to 240 minutes of audio, with each city providing 144 segments. The dataset provides no anomalous event. The TAU dataset was intended for the scene classification task and only provides the usual audio signals describing the scenes. We thus need to inject anomaly events within the regular audio. Therefore, we inject all three rare events from the TUT dataset, namely *babycry*, *gunshot*, and *glass-break*. We create three separate test sets to investigate the role of the audio signature of the anomalous events in its detection.

The employed injection procedure is similar to that proposed in Mesaros et al. (2017) for the TUT dataset. However, we drastically increased the event occurrence probability bringing it to the 60% of the chance of having an injection[7]. That contrasts with the previous dataset in which the number of anomaly events is relatively small. Here, the high number of corrupted samples should enable us to cover a wider variety of mixtures between background sounds and anomalous events, making the analysis more accurate. Finally, we mix the background audio signal and the anomalous event at three different levels of event-to-background ratios[8](EBR) of $\{-6dB, +0dB, +6dB\}$.

[5] Even with a large amount of GPU RAM(32GB) we were not able to train the model so as it was provided from the authors of (Rushe et al., 2019).

[6] http://dcase.community/challenge2019/task-acoustic-scene-classification.

[7] In the previous experiments the ratio ranges between around the 2% and 5% depending on the segment length. More details about this data distribution at AudioAnomalyDetectionWaveNet

[8] The event-to-background ratio is defined as the ratio between the root mean square energy of the rare event and the background audio.

Table 4 AUC and AUC PR results for SPAD at different time scales: 1s and 2s

Input Length	4096 - 16384 Samples		44100 Samples						88200 Samples					
Model	best-SPAD ($< 1s$)		fully-SPAD+CNN14		fully-SPAD		conv-SPAD PR		fully-SPAD+CNN14		fully-SPAD		conv-SPAD	
Place	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR	AUC	AUC PR
beach	0.85	0.54	0.86	0.36	**0.89**	0.54	**0.89**	**0.64**	0.86	0.37	**0.89**	0.60	0.83	0.49
bus	0.89	0.67	0.89	0.55	0.90	0.71	0.87	0.67	0.89	0.65	0.90	**0.75**	**0.91**	0.65
cafe restaurant	0.76	0.34	0.75	0.27	**0.80**	0.36	0.78	**0.48**	0.74	0.35	0.78	0.27	0.71	0.24
car	0.91	0.70	0.91	0.58	0.92	0.73	0.92	0.73	0.89	0.64	**0.93**	**0.79**	**0.93**	0.78
city center	0.87	0.56	0.87	0.61	0.89	0.55	0.70	0.12	0.91	**0.69**	**0.92**	0.66	0.88	0.52
forest path	0.79	0.20	**0.82**	0.18	**0.82**	0.21	0.76	0.17	0.80	**0.24**	0.78	**0.24**	0.75	0.19
grocery store	0.84	0.50	0.83	0.35	0.86	0.42	0.86	**0.52**	0.88	0.42	**0.91**	0.45	0.88	0.41
home	**0.72**	**0.15**	**0.72**	0.13	0.70	0.07	**0.72**	0.09	0.71	0.12	0.64	0.07	0.66	0.09
library	0.75	0.24	0.80	0.37	0.73	0.12	0.79	0.14	**0.85**	**0.45**	0.76	0.12	0.65	0.11
metro station	0.83	0.42	0.82	0.37	0.85	0.41	0.85	0.41	0.86	**0.45**	**0.89**	0.44	0.88	0.37
office	0.83	0.44	0.84	0.43	0.84	0.37	0.84	0.44	**0.87**	**0.57**	0.86	0.37	0.79	0.21
park	0.85	0.53	0.87	0.34	0.87	0.52	0.87	0.57	0.88	0.39	**0.89**	**0.59**	0.72	0.40
residential area	0.82	0.39	0.83	0.39	0.86	0.38	0.84	0.43	0.82	0.39	**0.89**	**0.45**	0.65	0.08
train	0.86	0.53	**0.88**	0.44	**0.88**	**0.58**	0.86	0.56	**0.88**	0.44	**0.88**	**0.58**	**0.88**	0.53
tram	0.87	0.66	0.87	0.63	0.90	0.73	0.90	0.74	0.93	0.77	**0.94**	**0.81**	0.79	0.43
Overall Score	0.83	0.46	0.84	0.4	0.85	0.45	0.83	0.45	0.85	0.46	**0.86**	**0.48**	0.79	0.37

The best results are marked in bold

Table 5 TAU Dataset: comparative experiment between Conv-SPAD and WaveNet with input length 4096 sample

Place	Metric	Model	-6 dB	0 dB	6 dB
Babycry					
airport	AUC	Conv-SPAD	**0.889**	**0.859**	**0.833**
		WaveNet	0.402	0.563	0.703
	AUC PR	Conv-SPAD	**0.913**	**0.894**	**0.879**
		WaveNet	0.444	0.583	0.724
bus	AUC	Conv-SPAD	0.686	0.748	0.853
		WaveNet	**0.835**	**0.924**	**0.965**
	AUC PR	Conv-SPAD	0.783	0.834	0.914
		WaveNet	**0.839**	**0.929**	**0.968**
metro	AUC	Conv-SPAD	0.721	0.711	0.764
		WaveNet	**0.735**	**0.854**	**0.919**
	AUC PR	Conv-SPAD	**0.800**	0.792	0.842
		WaveNet	0.732	**0.858**	**0.925**
metro station	AUC	Conv-SPAD	**0.864**	**0.869**	**0.872**
		WaveNet	0.577	0.706	0.798
	AUC PR	Conv-SPAD	**0.906**	**0.912**	**0.917**
		WaveNet	0.541	0.658	0.762
park	AUC	Conv-SPAD	**0.845**	**0.841**	0.850
		WaveNet	0.647	0.783	**0.875**
public square	AUC	Conv-SPAD	**0.664**	0.667	0.723
		WaveNet	0.550	**0.687**	**0.790**
	AUC PR	Conv-SPAD	**0.745**	**0.742**	**0.786**
		WaveNet	0.531	0.655	0.769
shopping mall	AUC	Conv-SPAD	**0.855**	**0.869**	0.867
		WaveNet	0.379	0.569	0.724
	AUC PR	Conv-SPAD	**0.913**	**0.905**	**0.908**
		WaveNet	0.465	0.630	0.773
street pedestrian	AUC	Conv-SPAD	**0.799**	**0.789**	0.802
		WaveNet	0.452	0.610	0.741
	AUC PR	Conv-SPAD	**0.861**	**0.852**	**0.862**
		WaveNet	0.484	0.636	0.768
street traffic	AUC	Conv-SPAD	**0.880**	**0.885**	0.895
		WaveNet	0.562	0.703	0.809
	AUC PR	Conv-SPAD	**0.917**	**0.920**	**0.928**
		WaveNet	0.554	0.701	0.815
tram	AUC	Conv-SPAD	0.734	0.729	0.785
		WaveNet	**0.738**	**0.849**	**0.914**
	AUC PR	Conv-SPAD	**0.813**	0.809	0.855
		WaveNet	0.717	**0.843**	0.915
Overall Score	AUC	Conv-SPAD	**0.797**	**0.797**	**0.824**
		WaveNet	0.588	0.725	**0.824**

Table 5 continued

	AUC PR	Conv-SPAD	**0.854**	**0.855**	0.879
		WaveNet	0. 594	0.727	0.829
Glassbreak					
airport	AUC	Conv-SPAD	**0.925**	**0.925**	0.873
		WaveNet	0.739	0.857	**0.919**
	AUC PR	Conv-SPAD	**0.946**	**0.939**	0.922
		WaveNet	0.766	0.881	**0.936**
bus	AUC	Conv-SPAD	0.734	0.719	0.769
		WaveNet	**0.956**	**0.980**	**0.990**
	AUC PR	Conv-SPAD	0.841	0.838	0.876
		WaveNet	**0.962**	**0.983**	**0.992**
metro	AUC	Conv-SPAD	0.744	0.745	0.747
		WaveNet	**0.921**	**0.961**	**0.979**
	AUC PR	Conv-SPAD	0.859	0.847	0.858
		WaveNet	**0.931**	**0.968**	**0.983**
metro station	AUC	Conv-SPAD	**0.876**	0.866	0.847
		WaveNet	0.819	**0.893**	**0.933**
	AUC PR	Conv-SPAD	0.923	**0.920**	0.911
		WaveNet	0.800	0.886	**0.933**
park	AUC	Conv-SPAD	**0.868**	0.845	0.824
		WaveNet	0.866	**0.932**	**0.964**
	AUC PR	Conv-SPAD	0.922	0.910	0.902
		WaveNet	0.877	**0.941**	**0.970**
public square	AUC	Conv-SPAD	0.738	0.733	0.734
		WaveNet	**0.813**	**0.893**	**0.936**
	AUC PR	Conv-SPAD	**0.823**	0.824	0.834
		WaveNet	0.798	**0.890**	**0.940**
shopping mall	AUC	Conv-SPAD	**0.919**	**0.907**	0.882
		WaveNet	0.786	0.893	**0.941**
	AUC PR	Conv-SPAD	**0.944**	**0.941**	0.931
		WaveNet	0.831	0.920	**0.958**
street pedestrain	AUC	Conv-SPAD	**0.841**	0.828	0.808
		WaveNet	0.763	**0.871**	**0.928**
	AUC PR	Conv-SPAD	**0.900**	**0.894**	0.888
		WaveNet	0.794	0.896	**0.945**
street traffic	AUC	Conv-SPAD	**0.909**	0.904	0.889
		WaveNet	0.844	**0.918**	**0.953**
	AUC PR	Conv-SPAD	**0.943**	**0.943**	0.936
		WaveNet	0.856	0.929	**0.961**
tram	AUC	Conv-SPAD	0.810	0.774	0.764
		WaveNet	**0.913**	**0.956**	**0.976**
Overall Score	AUC	Conv-SPAD	0.893	0.823	0.814
		WaveNet	**0.842**	**0.915**	**0.952**

Table 5 continued

	AUC PR	Conv-SPAD	**0.898**	0.892	0.892
		WaveNet	0.851	**0.926**	**0.960**
Gunshot					
airport	AUC	Conv-SPAD	**0.884**	**0.828**	**0.736**
		WaveNet	0.332	0.423	0.516
	AUC PR	Conv-SPAD	**0.913**	**0.874**	**0.812**
		WaveNet	0.462	0.534	0.612
bus	AUC	Conv-SPAD	0.686	0.748	0.853
		WaveNet	**0.835**	**0.924**	**0.965**
	AUC PR	Conv-SPAD	0.783	0.834	0.914
		WaveNet	**0.839**	**0.929**	**0.968**
bus	AUC	Conv-SPAD	0.590	0.586	0.708
		WaveNet	**0.686**	**0.796**	**0.871**
	AUC PR	Conv-SPAD	**0.722**	0.722	0.822
		WaveNet	**0.722**	**0.823**	**0.891**
metro	AUC	Conv-SPAD	**0.658**	0.584	0.585
		WaveNet	0.581	**0.693**	**0.778**
	AUC PR	Conv-SPAD	**0.765**	0.771	0.713
		WaveNet	0.638	**0.737**	**0.813**
metro station	AUC	Conv-SPAD	**0.755**	**0.714**	**0.671**
		WaveNet	0.475	0.567	0.647
	AUC PR	Conv-SPAD	**0.831**	**0.802**	**0.770**
		WaveNet	0.528	0.602	0.673
park	AUC	Conv-SPAD	**0.777**	**0.731**	0.715
		WaveNet	0.520	0.631	0.730
	AUC PR	Conv-SPAD	**0.851**	**0.812**	**0.805**
		WaveNet	0.585	0.681	0.766
public square	AUC	Conv-SPAD	**0.631**	**0.571**	0.555
		WaveNet	0.452	0.544	**0.631**
	AUC PR	Conv-SPAD	**0.736**	**0.690**	**0.680**
		WaveNet	0.523	0.597	0.671
shopping mall	AUC	Conv-SPAD	**0.870**	**0.819**	**0.746**
		WaveNet	0.304	0.403	0.507
	AUC PR	Conv-SPAD	**0.905**	**0.870**	**0.820**
		WaveNet	0.477	0.555	0.636
street pedestrian	AUC	Conv-SPAD	**0.769**	**0.712**	**0.665**
		WaveNet	0.368	0.463	0.556
	AUC PR	Conv-SPAD	**0.847**	**0.805**	**0.768**
		WaveNet	0.484	0.563	0.643
street traffic	AUC	Conv-SPAD	**0.823**	**0.770**	**0.726**
		WaveNet	0.465	0.558	0.642
	AUC PR	Conv-SPAD	**0.879**	**0.841**	**0.805**
		WaveNet	0.543	0.621	0.695

Table 5 continued

tram	AUC	Conv-SPAD	**0.687**	0.605	0.604
		WaveNet	0.589	**0.698**	**0.782**
	AUC PR	Conv-SPAD	**0.787**	0.729	0.730
		WaveNet	0.611	**0.730**	**0.808**
Overall score	AUC	Conv-SPAD	**0.744**	**0.692**	**0.671**
		WaveNet	0.477	0.578	0.666
	AUC PR	Conv-SPAD	**0.823**	**0.786**	**0.773**
		WaveNet	0.557	0.644	0.721

The best results are marked in bold

As in the previous experiments, we use two different chuck sizes, 4096 and 16384 audio samples, for the comparative analysis against WaveNet and 44100 and 88200 audio samples only for the Conv-Spad model.

Analyzing the results in Table 5, we can confirm the competitive performance of the simple and lightweight model conv-Spad. With the input size of 4096 audio samples, the 155k parameters of the conv-Spad are enough to reach comparable results with the larger model WaveNet. Looking at the aggregate score conv-Spad can outperform WaveNet five times over nine and match on one when considering AUC. At the same time, it has a score of six to three over the AUC PR. The experiments also highlight how the deeper sounds of the gunshot are more challenging to be detected than babycry and glass-break in the frequency domain. That happens since the event gunshot has a lower frequency content that is more prone to overlap with background sound frequencies, especially at higher EBR levels and in short audio segments. On the other hand, the audio signatures of babycry and glass-break, characterized by a sharp energy content mainly concentrating at higher frequencies, make these anomalous events stand out even at low EBR. The same consideration does not apply to the waveform analysis made by WaveNet, which on the other hand, takes advantage of that.

The results for the input size of 16384 audio samples, Table 6 align with the previous and show how the small model conv-Spad, with its 477k parameters, can outperform WaveNet seven times out of nine both over the AUC and AUC PR. Furthermore, the improvements in the results follow the trend already seen with the TUT dataset, in which broader inputs enable the model to reach better results. To confirm this last point, we present the results for the largest input size we tested, 44100 and 88200 audio samples.

We observe a substantial performance increment in the results reported in Table 7. Furthermore, conv-Spad performs better with larger than smaller inputs over all the different types of anomaly events. That is related to the spectral analysis made on the input signal. When we perform such analysis over small inputs, we are forced to shrink the window size for the spectrogram computation. For example, for input length 4096, we set the window size to 256 points, and the spectrum is equally split into images representing a 5ms duration[9]. However, reducing the window size increases the lowest detectable frequency, $F_0 = 5 * (SR/WS) = 5 * (44100/265) \simeq 861Hz$. In Table 7, we use a 1024 samples analysis window that allows us to capture lower pitch signals, $F_0 \simeq 215Hz$. From the result,

[9] When we compute the spectrogram, we use a window size of 256 points and hope size of 80 points for audio segments shorter than 44100 samples, 1024 and 320 for equal to or longer segments.

Table 6 TAU Dataset: comparative experiment between Conv-SPAD and WaveNet with input length 16384 samples

Place	Metric	Model	-6 dB	0 dB	6 dB
Babycry					
airport	AUC	Conv-SPAD	**0.903**	**0.884**	**0.873**
		WaveNet	0.349	0.509	0.664
	AUC PR	Conv-SPAD	**0.930**	**0.916**	**0.908**
		WaveNet	0.512	0.631	0.762
bus	AUC	Conv-SPAD	**0.921**	0.935	0.943
		WaveNet	0.883	**0.947**	**0.977**
	AUC PR	Conv-SPAD	**0.934**	0.951	0.961
		WaveNet	0.906	**0.961**	**0.984**
metro	AUC	Conv-SPAD	**0.854**	0.813	0.793
		WaveNet	0.760	**0.871**	**0.932**
	AUC PR	Conv-SPAD	**0.898**	0.870	0.855
		WaveNet	0.812	**0.908**	**0.955**
metro station	AUC	Conv-SPAD	**0.894**	**0.931**	**0.949**
		WaveNet	0.557	0.683	0.778
	AUC PR	Conv-SPAD	**0.921**	**0.953**	**0.968**
		WaveNet	0.612	0.702	0.781
park	AUC	Conv-SPAD	**0.957**	**0.972**	**0.977**
		WaveNet	0.663	0.800	0.888
	AUC PR	Conv-SPAD	**0.972**	**0.984**	**0.987**
		WaveNet	0.726	0.843	0.917
park	AUC	Conv-SPAD	**0.868**	**0.923**	**0.948**
		WaveNet	0.533	0.674	0.785
	AUC PR	Conv-SPAD	**0.912**	**0.953**	**0.971**
		WaveNet	0.610	0.718	0.816
shopping mall	AUC	Conv-SPAD	**0.926**	**0.921**	**0.927**
		WaveNet	0.334	0.530	0.704
	AUC PR	Conv-SPAD	**0.945**	**0.942**	**0.947**
		WaveNet	0.527	0.679	0.813
street pedestrian	AUC	Conv-SPAD	**0.860**	**0.872**	**0.893**
		WaveNet	0.428	0.591	0.734
	AUC PR	Conv-SPAD	**0.911**	**0.915**	**0.928**
		WaveNet	0.563	0.701	0.820
street traffic	AUC	Conv-SPAD	**0.940**	**0.968**	**0.978**
		WaveNet	0.550	0.694	0.807
	AUC PR	Conv-SPAD	**0.959**	**0.981**	**0.988**
		WaveNet	0.529	0.677	0.803
Overall Score	AUC	Conv-SPAD	**0.904**	**0.915**	**0.922**
		WaveNet	0.582	0.716	0.819
	AUC PR	Conv-SPAD	**0.933**	**0.942**	**0.948**
		WaveNet	0.659	0.771	0.859

Table 6 continued

Glassbreak					
airport	AUC	Conv-SPAD	**0.933**	**0.912**	0.874
		WaveNet	0.734	0.858	**0.925**
	AUC PR	Conv-SPAD	**0.955**	**0.942**	0.924
		WaveNet	0.821	0.913	**0.957**
bus	AUC	Conv-SPAD	0.850	0.860	0.866
		WaveNet	**0.971**	**0.987**	**0.994**
	AUC PR	Conv-SPAD	0.895	0.907	0.915
		WaveNet	**0.981**	**0.992**	**0.996**
metro	AUC	Conv-SPAD	0.877	0.813	0.746
		WaveNet	**0.939**	**0.972**	**0.985**
	AUC PR	Conv-SPAD	0.918	0.882	0.845
		WaveNet	**0.962**	**0.984**	**0.992**
metro station	AUC	Conv-SPAD	**0.860**	0.876	0.885
		WaveNet	0.817	**0.885**	**0.924**
	AUC PR	Conv-SPAD	**0.913**	**0.927**	**0.936**
		WaveNet	0.810	0.870	0.906
park	AUC	Conv-SPAD	**0.897**	0.914	0.921
		WaveNet	0.887	**0.944**	**0.973**
	AUC PR	Conv-SPAD	**0.936**	0.950	0.956
		WaveNet	0.923	**0.965**	**0.984**
public square	AUC	Conv-SPAD	**0.835**	0.859	0.874
		WaveNet	0.820	**0.898**	**0.941**
	AUC PR	Conv-SPAD	**0.903**	**0.921**	0.933
		WaveNet	0.853	**0.921**	**0.958**
shopping mall	AUC	Conv-SPAD	**0.950**	**0.937**	**0.917**
		WaveNet	0.788	0.898	0.949
	AUC PR	Conv-SPAD	**0.965**	**0.959**	**0.950**
		WaveNet	0.875	0.943	0.973
street pedestrian	AUC	Conv-SPAD	**0.888**	0.877	0.863
		WaveNet	0.781	**0.886**	**0.941**
	AUC PR	Conv-SPAD	**0.934**	0.928	0.922
		WaveNet	0.859	**0.933**	**0.967**
street traffic	AUC	Conv-SPAD	**0.895**	0.917	0.932
		WaveNet	0.861	**0.930**	**0.962**
	AUC PR	Conv-SPAD	**0.938**	**0.953**	**0.963**
		WaveNet	0.840	0.910	0.940
tram	AUC	Conv-SPAD	0.835	0.842	0.844
		WaveNet	**0.929**	**0.965**	**0.982**
	AUC PR	Conv-SPAD	0.903	0.912	0.917
		WaveNet	**0.951**	**0.978**	**0.989**
Overall Score	AUC	Conv-SPAD	**0.882**	0.881	0.872
		WaveNet	0.853	**0.922**	**0.958**

Table 6 continued

	AUC PR	Conv-SPAD	**0.926**	0.928	0.926
		WaveNet	0.888	**0.941**	**0.966**
Gunshot					
airport	AUC	Conv-SPAD	**0.897**	**0.848**	**0.777**
		WaveNet	0.292	0.378	0.469
	AUC PR	Conv-SPAD	**0.930**	**0.897**	**0.851**
		WaveNet	0.529	0.594	0.665
bus	AUC	Conv-SPAD	0.603	0.629	0.657
		WaveNet	**0.743**	**0.839**	**0.898**
	AUC PR	Conv-SPAD	0.693	0.717	0.745
		WaveNet	**0.819**	**0.891**	**0.935**
metro	AUC	Conv-SPAD	**0.836**	**0.753**	0.669
		WaveNet	0.597	0.709	**0.791**
	AUC PR	Conv-SPAD	**0.884**	**0.829**	0.772
		WaveNet	0.725	0.808	**0.868**
metro station	AUC	Conv-SPAD	**0.588**	**0.602**	0.624
		WaveNet	0.461	0.548	**0.627**
	AUC PR	Conv-SPAD	**0.696**	**0.718**	**0.744**
		WaveNet	0.584	0.645	0.707
park	AUC	Conv-SPAD	textbf0.755	**0.797**	**0.821**
		WaveNet	0.513	0.635	0.738
	AUC PR	Conv-SPAD	**0.815**	**0.858**	**0.882**
		WaveNet	0.668	0.755	0.828
public square	AUC	Conv-SPAD	**0.547**	**0.564**	0.587
		WaveNet	0.439	0.530	**0.617**
	AUC PR	Conv-SPAD	**0.672**	**0.695**	0.724
		WaveNet	0.599	0.666	**0.732**
shopping mall	AUC	Conv-SPAD	**0.916**	**0.881**	**0.845**
		WaveNet	0.270	0.369	0.473
	AUC PR	Conv-SPAD	**0.940**	**0.917**	**0.892**
		WaveNet	0.539	0.615	0.691
street pedestrian	AUC	Conv-SPAD	**0.823**	**0.787**	**0.767**
		WaveNet	0.348	0.443	0.538
	AUC PR	Conv-SPAD	**0.893**	**0.867**	**0.848**
		WaveNet	0.560	0.634	0.708
street traffic	AUC	Conv-SPAD	**0.706**	**0.752**	**0.800**
		WaveNet	0.460	0.551	0.635
	AUC PR	Conv-SPAD	**0.787**	**0.828**	**0.867**
		WaveNet	0.509	0.590	0.675
tram	AUC	Conv-SPAD	0.544	0.554	0.578
		WaveNet	**0.616**	**0.718**	**0.795**
	AUC PR	Conv-SPAD	0.668	0.686	0.716
		WaveNet	**0.720**	**0.800**	**0.861**

Table 6 continued

Overall Score	AUC	Conv-SPAD	0.721	0.717	0.713
		WaveNet	**0.721**	**0.717**	**0.713**
	AUC PR	Conv-SPAD	**0.798**	**0.801**	**0.804**
		WaveNet	0.625	0.700	0.767

The best results are marked in bold

we can see how those lower frequencies enable the model to capture more details about the place's original signature, making the anomaly detection task more effective. Once again, we recall that we were not able to test WaveNet with these input lengths since the model training was excessively memory-demanding. Finally, the results show how conv-SPAD over 1s input is a good trade-off solution for anomaly detection on edge devices. Only 1.2M parameters make our proposed architecture, and its extremely simple layers can be replaced by more sophisticated layers and further reduce its size. Also, we plan to investigate recent approaches for quantization and distillation in future works.

6 Conclusion

In this work, we conducted an analysis to design a suitable lightweight model to perform audio-based anomaly detection on edge devices. In particular, we investigated the performance of solutions built upon an autoencoder architecture and the spectral analysis of the input audio signal. We evaluated the role played by general-purpose audio features extracted via a transfer learning approach and shown that the same performance could be reached by increasing the capacity of a fully connected autoencoder. We then introduced conv-SPAD, a lightweight convolutional autoencoder that can heavily reduce the model's memory footprint without excessively sacrificing the performance. With conv-SPAD we can show that even a simple architecture can beat a larger model, such as WaveNet, in the task of anomaly detection across different real scenarios. Moreover, its small computational cost allows this model to be used on edge devices. From the experiments, we can thus conclude that, as a model suitable for edge devices, the conv-SPAD model results in a competitive approach to carry out audio-based anomaly detection with inputs whose duration is at most one second. Of course, this latter point should guide the choice of this approach for compatible scenarios. For instance, for natural scenes, the structure of the typical input signature may be well captured by the model, even with a small input duration. Moreover, the limited input duration does not prevent the model from being sensible to anomalous events, even if a more prolonged duration better characterizes those events. That might not be true in scenarios such as industrial machine failures where the anomalies could be defined as altered frequency patterns that can only be detected by evaluating the temporal component, thus using some auto-regressive or recurrent approach. Nevertheless, unfortunately, those models are difficult to match the constraints of edge devices. In future work, we plan to extend our analysis to different application domains (e.g. factory machine failures) and different architectures. On this last point, we will consider to balance simplicity with regularized models ranging from variational approaches to more recent adversarial-based learning, such as Adversarial Autoencoders (Makhzani et al., 2016), that enable to use a broader set of distributions as priors for the latent code.

Table 7 TAU Dataset: comparative experiment on Conv-SPAD with input length 44100 and 88200 samples

Place	Metric	Segment Length	-6 dB	0 dB	6 dB
Babycry					
airport	AUC	44100	0.923	0.963	0.979
		88200	0.917	0.961	0.980
	AUC PR	44100	0.922	0.966	0.984
		88200	0.907	0.961	0.985
bus	AUC	44100	0.961	0.972	0.977
		88200	0.970	0.981	0.986
	AUC PR	44100	0.966	0.978	0.984
		88200	0.974	0.985	0.990
metro	AUC	44100	0.950	0.971	0.979
		88200	0.943	0.975	0.986
	AUC PR	44100	0.966	0.982	0.988
		88200	0.963	0.985	0.992
metro station	AUC	44100	0.944	0.975	0.986
		88200	0.932	0.975	0.990
	AUC PR	44100	0.959	0.983	0.992
		88200	0.951	0.983	0.994
park	AUC	44100	0.983	0.992	0.994
		88200	0.973	0.990	0.995
	AUC PR	44100	0.987	0.994	0.996
		88200	0.978	0.993	0.997
public square	AUC	44100	0.925	0.968	0.983
		88200	0.924	0.971	0.987
	AUC PR	44100	0.947	0.979	0.990
		88200	0.943	0.979	0.991
shopping mall	AUC	44100	0.945	0.980	0.990
		88200	0.952	0.985	0.994
	AUC PR	44100	0.955	0.986	0.994
		88200	0.960	0.989	0.996
street pedestrian	AUC	44100	0.943	0.975	0.986
		88200	0.938	0.978	0.990
	AUC PR	44100	0.956	0.984	0.992
		88200	0.951	0.984	0.994
street traffic	AUC	44100	0.978	0.992	0.996
		88200	0.982	0.995	0.997
	AUC PR	44100	0.984	0.995	0.998
		88200	0.987	0.996	0.998
tram	AUC	44100	0.953	0.970	0.975
		88200	0.957	0.976	0.983
	AUC PR	44100	0.967	0.981	0.986
		88200	0.970	0.985	0.990

Table 7 continued

Overall Score	AUC	44100	0.950	0.976	0.985
		88200	0.949	0.979	0.989
	AUC PR	44100	0.961	0.983	0.990
		88200	0.958	0.984	0.993
Glassbreak					
airport	AUC	44100	0.887	0.917	0.935
		88200	0.872	0.918	0.947
	AUC PR	44100	0.911	0.939	0.957
		88200	0.867	0.912	0.945
bus	AUC	44100	0.897	0.911	0.920
		88200	0.912	0.935	0.951
	AUC PR	44100	0.918	0.934	0.947
		88200	0.916	0.943	0.962
metro	AUC	44100	0.887	0.904	0.914
		88200	0.879	0.915	0.940
	AUC PR	44100	0.927	0.943	0.953
		88200	0.913	0.944	0.964
metro station	AUC	44100	0.909	0.931	0.943
		88200	0.912	0.943	0.962
	AUC PR	44100	0.937	0.955	0.966
		88200	0.934	0.960	0.976
park	AUC	44100	0.951	0.964	0.968
		88200	0.950	0.970	0.980
	AUC PR	44100	0.968	0.978	0.982
		88200	0.959	0.978	0.987
public square	AUC	44100	0.880	0.914	0.933
		88200	0.871	0.922	0.955
	AUC PR	44100	0.920	0.947	0.963
		88200	0.893	0.940	0.969
shopping mall	AUC	44100	0.917	0.942	0.957
		88200	0.934	0.962	0.977
	AUC PR	44100	0.944	0.964	0.976
		88200	0.943	0.969	0.984
street pedestrian	AUC	44100	0.898	0.926	0.941
		88200	0.892	0.934	0.960
	AUC PR	44100	0.930	0.954	0.967
		88200	0.906	0.947	0.971
street traffic	AUC	44100	0.966	0.977	0.982
		88200	0.983	0.989	0.991
	AUC PR	44100	0.979	0.986	0.989
		88200	0.988	0.993	0.995
tram	AUC	44100	0.892	0.907	0.913
		88200	0.883	0.918	0.941

Table 7 continued

	AUC PR	44100	0.927	0.944	0.953
		88200	0.901	0.938	0.962
Overall Score	AUC	44100	0.909	0.929	0.941
		88200	0.909	0.941	0.960
	AUC PR	44100	0.936	0.954	0.965
		88200	0.922	0.952	0.972
Gunshot					
airport	AUC	44100	0.637	0.673	0.721
		88200	0.654	0.712	0.777
	AUC PR	44100	0.712	0.748	0.796
		88200	0.702	0.751	0.809
bus	AUC	44100	0.689	0.731	0.774
		88200	0.739	0.784	0.825
	AUC PR	44100	0.748	0.798	0.838
		88200	0.781	0.826	0.869
metro	AUC	44100	0.629	0.653	0.694
		88200	0.627	0.677	0.732
	AUC PR	44100	0.731	0.763	0.803
		88200	0.733	0.781	0.828
metro station	AUC	44100	0.716	0.757	0.799
		88200	0.730	0.787	0.839
	AUC PR	44100	0.785	0.826	0.865
		88200	0.801	0.852	0.894
park	AUC	44100	0.824	0.868	0.894
		88200	0.835	0.886	0.917
	AUC PR	44100	0.867	0.908	0.932
		88200	0.866	0.915	0.944
public square	AUC	44100	0.639	0.681	0.732
		88200	0.674	0.739	0.802
	AUC PR	44100	0.722	0.770	0.823
		88200	0.729	0.792	0.857
shopping mall	AUC	44100	0.678	0.715	0.772
		88200	0.740	0.799	0.857
	AUC PR	44100	0.752	0.793	0.845
		88200	0.791	0.848	0.901
street pedestrian	AUC	44100	0.656	0.694	0.748
		88200	0.693	0.755	0.820
	AUC PR	44100	0.732	0.774	0.825
		88200	0.745	0.807	0.869
street traffic	AUC	44100	0.888	0.925	0.951
		88200	0.926	0.949	0.966
	AUC PR	44100	0.923	0.953	0.971
		88200	0.950	0.969	0.980

Table 7 continued

tram	AUC	44100	0.644	0.670	0.701
		88200	0.676	0.717	0.761
	AUC PR	44100	0.724	0.761	0.804
		88200	0.737	0.781	0.829
Overall Score	AUC	44100	0.700	0.737	0.779
		88200	0.729	0.780	0.830
	AUC PR	44100	0.770	0.809	0.850
		88200	0.784	0.832	0.878

The best results are marked in bold

Author Contributions Fabrizio Lo Scudo and Ettore Ritacco designed the architectures and the experiments. Fabrizio Lo Scudo implemented all models and carried out all the experimental tests, he also wrote the manuscript. Ettore Ritacco, Luciano Caroprese and Giuseppe Manco supervised the work and helped revise the manuscript. All authors approved the final version of the manuscript

Funding This work has been partially supported by MISE - PON Fabbrica Intelligente I&C 2014 - 2020 under project True Detective 4.0 N.PROG: F/190105/02/X44. Open access funding provided by Università della Calabria within the CRUI-CARE Agreement.

Availability of data and materials The datasets used in this work can be accessed at the following links. *TUT Rare Sound Events 2017* at https://zenodo.org/record/401395 and https://zenodo.org/record/1160455 *TAU Urban Acoustic Scenes 2020* datasets at https://zenodo.org/record/3670167

Declarations

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

Ethical Approval This research did not contain any studies involving animal or human participants, nor did it take place on any private or protected areas. No specific permissions were required for corresponding locations

References

Baldi, P. (2012) Autoencoders, unsupervised learning, and deep architectures. In: Guyon I, Dror G, Lemaire V, et al (eds) *Proceedings of ICML Workshop on Unsupervised and Transfer Learning, Proceedings of Machine Learning Research*, vol 27. PMLR, Bellevue, Washington, USA, pp. 37–49 (2012). http://proceedings.mlr.press/v27/baldi12a/baldi12a.pdf

Bayram, B., Duman, TB., & Ince, G. (2021). Real time detection of acoustic anomalies in industrial processes using sequential autoencoders. *Expert Systems 38*(1).https://doi.org/10.1111/exsy.12564

Becker, P., Roth, C., & Roennau, A., et al. (2020). Acoustic anomaly detection in additive manufacturing with long short-term memory neural networks. In: *2020 IEEE 7th International Conference on Indus-*

trial Engineering and Applications (ICIEA), pp. 921–926, https://doi.org/10.1109/ICIEA49774.2020.9102002

Bigo, D., Carrera, S., & Hernanz, N., et al. (2013). National programmes for mass surveillance of personal data in eu member states and their compatibility with eu law. *General for Internal Policies of the Union*. https://doi.org/10.2861/48584

Brazil, T. (1995). Causal-convolution-a new method for the transient analysis of linear systems at microwave frequencies. *IEEE Transactions on Microwave Theory and Techniques, 43*(2), 315–323. https://doi.org/10.1109/22.348090

Caire, P., Moawad, A., & Efthymiou, V., et al. (2016). Privacy challenges in ambient intelligence systems. *Journal of Ambient Intelligence and Smart Environments, 8*(6), 619–644. https://doi.org/10.3233/AIS-160405

Chandola, V., Banerjee A., & Kumar, V. (2009) Anomaly detection A survey. *ACM Comput Surv 41*(3). https://doi.org/10.1145/1541880.1541882

Chen, J., Kam, AH., & Zhang, J., et al. (2005). Bathroom activity monitoring based on sound. In: *International Conference on Pervasive Computing*, Springer, pp. 47–61. https://doi.org/10.1007/11428572_4

Cho, K., van Merriënboer, B., & Gulcehre, C., et al. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179

Clavel, C., Ehrette, T., & Richard, G. (2005). Events detection for an audio-based surveillance system. In: *2005 IEEE International Conference on Multimedia and Expo*, pp. 1306–1309. https://doi.org/10.1109/ICME.2005.1521669

Crocco, M., Cristani, M., & Trucco, A., et al. (2016). Audio surveillance A systematic review. *ACM Comput Surv 48*(4). https://doi.org/10.1145/2871183

Duman, TB., Bayram, B., & İnce, G. (2019). Acoustic anomaly detection using convolutional autoencoders in industrial processes. In: *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019)*. Springer, pp. 432–442. https://doi.org/10.1007/978-3-030-20055-8_41.

EU-Agency (2017). Surveillance by intelligence services Fundamental rights safeguards and remedies in the eu. In: *Field Perspectives and Legal Update*, vol 2. Publications Office of the European Union Luxembourg, https://doi.org/10.2811/792946

Foggia, P., Petkov, N., & Saggese, A., et al. (2016). Audio surveillance of roads: A system for detecting anomalous sounds. *IEEE Transactions on Intelligent Transportation Systems, 17*(1), 279–288. https://doi.org/10.1109/TITS.2015.2470216

Gemmeke, JF., Ellis, DPW., & Freedman, D., et al. (2017). Audio set: An ontology and human-labeled dataset for audio events. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780. https://doi.org/10.1109/ICASSP.2017.7952261

Goetze, S., Moritz, N., & Appell, J. E., et al. (2010). Acoustic user interfaces for ambient-assisted living technologies. *Informatics for Health and Social Care, 35*(3–4), 125–143. https://doi.org/10.3109/17538157.2010.528655

Greengard, S. (2020). Ai on edge. *Commun ACM 63*(9). https://doi.org/10.1145/3409977

Haritaoglu, I., Harwood, D., & Davis, L. (2000). W/sup 4/: real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(8), 809–830. https://doi.org/10.1109/34.868683

Hayashi, T., Komatsu, T., Kondo, R., et al. (2018). Anomalous sound event detection based on wavenet. In: *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 2494–2498,https://doi.org/10.23919/EUSIPCO.2018.8553423

Hinton, G., Deng, L., & Yu, D., et al. (2012). Deep neural networks for acoustic modeling in speech recognition The shared views of four research groups. *IEEE Signal Processing Magazine, 29*(6), 82–97. https://doi.org/10.1109/MSP.2012.2205597

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial Intelligence Review, 22*(2), 85–126. https://doi.org/10.1007/s10462-004-4304-y

Imai, S. (1983). Cepstral analysis synthesis on the mel frequency scale. In: *ICASSP '83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 93–96. https://doi.org/10.1109/ICASSP.1983.1172250

Kawaguchi, Y., Tanabe, R., & Endo, T., et al. (2019). Anomaly detection based on an ensemble of dereverberation and anomalous sound extraction. In: *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 865–869, https://doi.org/10.1109/ICASSP.2019.8683702

Koizumi, Y., Murata, S., & Harada, N., et al. (2019a) Sniper: Few-shot learning for anomaly detection to minimize false-negative rate with ensured true-positive rate. In: *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 915–919, https://doi.org/10.1109/ICASSP.2019.8683667.

Koizumi, Y., Saito, & S., Uematsu H, et al. (2019b) Unsupervised detection of anomalous sound based on deep learning and the neyman – pearson lemma. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 27*(1), pp. 212–224. https://doi.org/10.1109/TASLP.2018.2877258.

Kong, Q., Cao, Y., Iqbal, T., et al. (2020). Panns Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28*, 2880–2894. https://doi.org/10.1109/TASLP.2020.3030497

Kuner, C. (2012). The european commission's proposed data protection regulation A copernican revolution in european data protection law. *Bloomberg BNA Privacy and Security Law Report 6*. http://arxiv.org/org/abs/ssrn.com/abstract=/2162781'.

Lee, YL., Tsung, PK., & Wu, M. (2018). Techology trend of edge ai. In: *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–2. https://doi.org/10.1109/VLSI-DAT.2018.8373244

Li, Z., Liu, F., Yang, W., et al. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems, 33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Makhzani, A., Shlens, J., & Jaitly, N., et al. (2016). Adversarial autoencoders. In: *International Conference on Learning Representations*, arXiv:1511.05644.

Marchi, E., Vesperini, F., & Eyben, F., et al. (2015a). A novel approach for automatic acoustic novelty detection using a denoising autoencoder with bidirectional lstm neural networks. In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1996–2000, https://doi.org/10.1109/ICASSP.2015.7178320

Marchi, E., Vesperini, F., & Weninger, F., et al. (2015b). Non-linear prediction with lstm recurrent neural networks for acoustic novelty detection. In: *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, https://doi.org/10.1109/IJCNN.2015.7280757.

Marchi, E., Vesperini, F., & Squartini, S., et al. (2017). Deep recurrent neural network-based autoencoders for acoustic novelty detection. *Computational Intelligence and Neuroscience*. https://doi.org/10.1155/2017/4694860

Mayer, R., & Jacobsen, HA. (2020). Scalable deep learning on distributed infrastructures Challenges, techniques, and tools. *ACM Comput Surv 53*(1). https://doi.org/10.1145/3363554

Mesaros, A., Heittola, T., & Diment, A., et al. (2017). DCASE 2017 Challenge setup Tasks, datasets and baseline system. In: *DCASE 2017 - Workshop on Detection and Classification of Acoustic Scenes and Events*, Munich, Germany, https://inria.hal.science/hal-01627981'.

Mesaros, A., Heittola, T., & Virtanen, T. (2018). A multi-device dataset for urban acoustic scene classification. In: *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, pp. 9–13. arXiv:1807.09840.

Neyman, J., & Pearson, ES. (1992). On the Problem of the Most Efficient Tests of Statistical Hypotheses, Springer, New York, New York, NY, pp. 73–108. https://doi.org/10.1007/978-1-4612-0919-5_6

Ntalampiras, S., Potamitis, I., & Fakotakis, N. (2009). On acoustic surveillance of hazardous situations. In: *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 165–168. https://doi.org/10.1109/ICASSP.2009.4959546

Ntalampiras, S., Potamitis, I., & Fakotakis, N. (2011). Probabilistic novelty detection for acoustic surveillance under real-world conditions. *IEEE Transactions on Multimedia, 13*(4), 713–719. https://doi.org/10.1109/TMM.2011.2122247

Nunes, EC. (2021). Anomalous sound detection with machine learning A systematic review. arXiv preprint arXiv:2102.07820.

van den Oord, A., Li, Y., & Babuschkin, I., et al. (2018). Parallel WaveNet: Fast high-fidelity speech synthesis. In: *Proceedings of the 35th International Conference on Machine Learning, Proceedings of Machine Learning Research*, vol. 80. PMLR, pp. 3918–3926. http://proceedings.mlr.press/v80/oord18a/oord18a.pdf'.

Oord Avd, Dieleman S, Zen H, et al (2016) Wavenet A generative model for raw audio. arXiv1609.03499 https://doi.org/10.48550.

Oruh, J., & Viriri, S. (2021). Spectral analysis for automatic speech recognition and enhancement. In: *Machine Learning for Networking Third International Conference*, MLN 2020, Paris, France, November 24–26,

2020, Revised Selected Papers 3, Springer International Publishing, pp. 245–254. https://doi.org/10.1007/978-3-030-70866-5_16

Paine, TL., Khorrami, P., & Chang, S., et al. (2016). Fast wavenet generation algorithm. arXiv preprint arXiv:1611.09482 https://doi.org/10.48550.

Pang, G., Shen, C., & Cao, L., et al. (2021). Deep learning for anomaly detection A review. *ACM Comput Surv 54*(2). https://doi.org/10.1145/3439950

Papadimitriou, I., Vafeiadis, A., & Lalas, A., et al. (2020). Audio-based event detection at different snr settings using two-dimensional spectrogram magnitude representations. *Electronics 9*(10). https://doi.org/10.3390/electronics9101593. www.mdpi.com/2079-9292/9/10/1593

Patcha, A., & Park, J. M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks, 51*(12), 3448–3470. https://doi.org/10.1016/j.comnet.2007.02.001

Rushe, E., & Namee, BM. (2019). Anomaly detection in raw audio using deep autoregressive networks. In: *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3597–3601.https://doi.org/10.1109/ICASSP.2019.8683414

Sánchez, J., Corral, G., & de Pozuelo, R. M., et al. (2016). Security issues and threats that may affect the hybrid cloud of finesce. *Netw Protoc Algorithms, 8*(1), 26–57. https://doi.org/10.5296/npa.v8i1.8727

Suefusa, K., Nishida, T., & Purohit, H., et al. (2020). Anomalous sound detection based on interpolation deep neural network. In: *ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 271–275. https://doi.org/10.1109/ICASSP40776.2020.9054344

Sutskever, I., Vinyals, O., & Le, Q.V. (2014). Sequence to sequence learning with neural networks. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, vol 2. MIT Press, Cambridge, MA, USA, pp. 3104—3112. https://doi.org/10.48550. arXiv:1409.3215

Theis, L., Shi, W., Cunningham, A., et al. (2017). Lossy image compression with compressive autoencoders. arXiv preprint arXiv:1703.00395 https://doi.org/10.48550

Valenzise, G., Gerosa, L., & Tagliasacchi, M., et al. (2007) Scream and gunshot detection and localization for audio-surveillance systems. In: *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 21–26. https://doi.org/10.1109/AVSS.2007.4425280.

Zhou, Z., Chen, X., Li, E., et al. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE, 107*(8), 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951