TYPE Original Research
PUBLISHED 12 July 2024
DOI 10.3389/frsip.2024.1358532

Check for updates

OPEN ACCESS

EDITED BY
Ina Kodrasi,
Idiap Research Institute, Switzerland

REVIEWED BY
Manuel Rosa Zurera,
University of Alcalá, Spain
Jaume Segura-Garcia,
University of Valencia, Spain

*CORRESPONDENCE
Stefano Damiano,
✉ stefano.damiano@esat.kuleuven.be

RECEIVED 19 December 2023
ACCEPTED 06 June 2024
PUBLISHED 12 July 2024

CITATION
Damiano S, Cramer B, Guntoro A and van Waterschoot T (2024), Synthetic data generation techniques for training deep acoustic siren identification networks. *Front. Sig. Proc.* 4:1358532. doi: 10.3389/frsip.2024.1358532

Synthetic data generation techniques for training deep acoustic siren identification networks

Stefano Damiano[1]*, Benjamin Cramer[2], Andre Guntoro[2] and Toon van Waterschoot[1]

[1]STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Department of Electrical Engineering (ESAT), KU Leuven, Leuven, Belgium, [2]Robert Bosch GmbH, Renningen, Germany

Acoustic sensing has been widely exploited for the early detection of harmful situations in urban environments: in particular, several siren identification algorithms based on deep neural networks have been developed and have proven robust to the noisy and non-stationary urban acoustic scene. Although high classification accuracy can be achieved when training and evaluating on the same dataset, the cross-dataset performance of such models remains unexplored. To build robust models that generalize well to unseen data, large datasets that capture the diversity of the target sounds are needed, whose collection is generally expensive and time consuming. To overcome this limitation, in this work we investigate synthetic data generation techniques for training siren identification models. To obtain siren source signals, we either collect from public sources a small set of stationary, recorded siren sounds, or generate them synthetically. We then simulate source motion, acoustic propagation and Doppler effect, and finally combine the resulting signal with background noise. This way, we build two synthetic datasets used to train three different convolutional neural networks, then tested on real-world datasets unseen during training. We show that the proposed training strategy based on the use of recorded source signals and synthetic acoustic propagation performs best. In particular, this method leads to models that exhibit a better generalization ability, as compared to training and evaluating in a cross-dataset setting. Moreover, the proposed method loosens the data collection requirement and is entirely built using publicly available resources.

KEYWORDS

synthetic data generation, acoustic simulation, moving sound sources, data augmentation, siren identification, convolutional neural networks

1 Introduction

The urban environment is characterized by a complex and dynamic acoustic scene where several agents produce overlapping sound events, some of which are artificially designed to alert humans of emergency situations that require their attention (Marchegiani and Fafoutis, 2022). The urban soundscape, therefore, contains information about the city itself. From the analysis of a recorded audio signal various details can be extracted, ranging from high level information such as a description of the recorded acoustic scene (e.g., whether it is a street, a station, a store etc.), to more fine-grained figures such as the weather condition or the time of the day when the recording was taken, and further to the identification of specific sounds appearing throughout the recording (e.g., the recognition of an alarm sound).

As smart vehicles start to populate the roads, major efforts are being devoted to enhance their perceptual abilities with the aim of improving their environmental awareness and strengthening their capacity to interact with other road agents. Autonomous cars, in fact, rely on information extracted in real time from multi-modal sensors to understand the environment, take driving decisions and interact with each other and with human drivers (Hussain and Zeadally, 2019).

In recent years, visual perception has been the primary research focus and it constitutes the main source of information for autonomous vehicles (Hussain and Zeadally, 2019). Nevertheless, the analysis of acoustic signals can provide important complementary information to enhance their environmental awareness. As a matter of fact, harmful situations are often announced by a sound event: emergency vehicles such as ambulances, police cars or fire trucks are equipped with sirens that announce their proximity and require cars to clear the way, whereas car horns can be used to quickly attract attention on an acute danger. Acoustic perception, moreover, provides specific advantages over vision: first, it is effective in situations where occlusions (e.g., caused by presence of buildings or other vehicles) or low-visibility conditions hinder the observation of visual cues. Second, it allows the detection of events that are not characterized by a corresponding visual signal, such as a car honking. Moreover, the lower dimensionality of acoustic signals enables a computationally efficient processing that best fits the deployment on embedded devices for real-time on-vehicle operation (Yin et al., 2023). Finally, in the safety-critical context of autonomous driving, the use of different sensing modalities to retrieve information serves as a mean to boost the reliability of the system. Every sensor mounted on the vehicle suffers in fact from specific limitations: visual sensors are hindered by low-visibility conditions, whereas acoustic sensors are impacted by strong background noise produced by wind, the ego-vehicle and the traffic background. For this reason, the use of multiple sensing modalities can compensate for the individual drawbacks to produce a more reliable prediction of events happening in the surrounding, making the use of audio analysis a useful resource for the overall performance of the system.

These considerations fueled the research on how to identify (and localize) emergency sounds in traffic scenarios (Tran and Tsai, 2020; Cantarini et al., 2021; Furletov et al., 2021; Sun et al., 2021; Marchegiani and Newman, 2022; Walden et al., 2022). In particular, several studies have targeted acoustic siren identification using both traditional signal processing (Meucci et al., 2008; Fazenda et al., 2009) and deep learning (Tran and Tsai, 2020; Cantarini et al., 2021; Marchegiani and Newman, 2022). The problem of siren identification is a sub-task of environmental sound classification (Piczak, 2015) (i.e., the identification of artificial or natural audio events different from speech or music) where the aim is to discriminate a siren signal from any other sound, generically labeled as background noise. This task has some application-specific challenges: first, emergency sirens have different patterns that can be grouped into two-tone, wail and yelp, each with different periodicity and time-frequency evolution (Marchegiani and Fafoutis, 2022). Not only the type of sirens varies among different countries, but within each siren type further variations can be observed in the time-frequency behavior. In other words, even between two instances of the same siren some variations may occur: in Germany, for instance, the lower fundamental frequency of the two-tone siren used can vary within the interval $[360, 630]$Hz, as regulated in DIN14610:2022-03 (2022). Furthermore, the strong and non-stationary noise that characterizes the urban acoustic scene constitutes a major challenge for siren identification algorithms.

Deep-learning techniques have been proven to achieve high classification accuracy in low signal-to-noise-ratio (SNR) conditions (Marchegiani and Newman, 2022). Several models based on convolutional neural networks (CNNs) have been proposed and achieve a high classification accuracy ($>95\%$). A large-scale CNN based on ensemble models is proposed in Tran and Tsai (2020) and achieves state-of-the-art accuracy, but suffers from a large model size that hinders its deployment on embedded devices. In Marchegiani and Newman (2022) a U-net (Ronneberger et al., 2015) is employed to perform noise reduction and improve robustness to low SNR levels, but still entails a large size and requires complex training strategies. On the other hand, small-scale models based on CNNs have been successfully used to identify country-specific instances of a siren sound (Cantarini et al., 2021; Cantarini et al., 2022).

One of the main challenges in the design of robust siren identification systems is the strong diversity of siren signals across the world, that requires algorithms to have a good generalization ability to unseen data. Although using large recorded datasets that capture the diversity of the sirens might benefit generalization capabilities, data collection is an expensive and time consuming procedure that has a limited scalability to systems that should be deployed in several different countries. Moreover, only a few datasets for siren identification are available, with a limited class diversity (Asif et al., 2022; Shah and Singh, 2023), far from representing all the existing siren variations.

Due to the strong diversity of siren sounds, in a real-world setting it is unlikely that the data seen during training is able to accurately represent all possible scenarios that might be faced by the model deployed on vehicles. For this reason, evaluating the model performance on the same dataset used for training, as is usually done in the acoustic siren identification literature, may lead to over-optimistic accuracy estimations. As a first contribution, in this work we analyze the performance of different siren identification models in a cross-dataset setting, i.e., when the dataset used for evaluation differs from the one used for training. Under the assumption that in realistic scenarios the data that will be seen at test time is unavailable (or only partially available) during the design and training phases, we then propose, as a second and main contribution, the use of synthetic data to train siren identification models: this technique, widely adopted in the image recognition domain (Jaderberg et al., 2014), obviates the need to collect real-world data for training purpose, speeding up the model design loop and cutting data recording and labeling costs. We evaluate two different data generation techniques: first, we generate synthetic siren signals (of two-tone, wail and yelp type) emitted by emergency vehicles, and simulate source motion and acoustic propagation using the *pyroadacoustics* simulator (Damiano and van Waterschoot, 2022). Alternatively, we collect a small amount of samples of stationary sirens (of the three types) from public databases, and feed them to pyroadacoustics to simulate acoustic propagation and Doppler effect. As a third contribution, we introduce several modifications to three state-of-the-art CNN models, in order to build an effective end-to-end siren identification system. In particular, we first enhance an existing siren classification network (Cantarini et al., 2021), by introducing batch normalization layers and dropout operations to boost generalization, and global average pooling to reduce the model complexity. We then adapt two BCResNet architectures (Kim et al., 2021), originally proposed for the acoustic scene classification task, to target siren identification. These architectures, to the best of the authors' knowledge, are adopted for the first time in this work for the siren identification task.

We train the CNN models on the two synthetically generated datasets, and evaluate them on three real-world (unseen) datasets to assess the effectiveness of the data generation procedure. We finally show that the proposed training technique, based on the use of recorded siren signals and synthetic acoustic propagation, leads to a higher performance compared to models trained on a real-world dataset and evaluated in a cross-dataset setting.

The rest of the paper is organized as follows. In Section 2 we discuss related works on siren identification based on CNN models. In Section 3 we describe the proposed synthetic data generation strategies and training procedures, and introduce the CNN architectures adopted to solve the classification task. We then evaluate the training strategies and the different models in Section 4, and draw conclusions in Section 5.

2 Related work

2.1 Siren identification methods

In recent years the problem of identifying sirens in an urban environment has been addressed by exploiting traditional signal processing approaches (Meucci et al., 2008; Fazenda et al., 2009) as well as machine learning (Schröder et al., 2013; Nandwana and Hasan, 2016; Carmel et al., 2017) and deep learning (Beritelli et al., 2006; Tran and Tsai, 2020; Cantarini et al., 2021; Furletov et al., 2021; Sharma et al., 2021; Sun et al., 2021; Cantarini et al., 2022; Marchegiani and Newman, 2022; Walden et al., 2022) techniques. Most state-of-the-art solutions rely on deep neural networks, due to their proven robustness to strong background noise and complex non-stationary acoustic scenes (Tran and Tsai, 2020; Marchegiani and Newman, 2022). The majority of these models apply image processing techniques to a time-frequency representation of the audio signal, adopting short-time Fourier transforms (STFTs), gammatonegrams or log-mel spectrograms as input features; others, instead, use the raw audio waveform (Beritelli et al., 2006; Furletov et al., 2021), or a combination of the two approaches (Tran and Tsai, 2020; Sun et al., 2021). In more detail, in Tran and Tsai (2020) a two-branch neural network is proposed for the classification of sirens (two-tone, wail and yelp) and vehicle horns. The first branch combines mel-frequency cepstral coefficients (MFCCs) and log-mel spectrograms extracted from 1-channel audio signals in a 2D image, processed via a 2D-CNN architecture, while the second one employs a 1D-CNN to automatically extract features from the raw audio waveform. The ensemble of these two networks achieves an accuracy of 98.24% in the classification task. The model is trained and evaluated on a dataset containing both public and internal data. In Marchegiani and Newman (2022) the classification of emergency sirens (two-tone, wail, yelp) and vehicle horns is tackled via a multi-task learning scheme: a U-net (Ronneberger et al., 2015) architecture is adopted to apply semantic segmentation to gammatonegram features extracted from 1- and 2-channel audio signals, with the goal of removing the background noise. Fully-connected layers are added at the output of the encoder part of the network to perform classification. The model is trained and evaluated on synthetic data: recorded siren sounds are fed to filters that simulate acoustic propagation and Doppler effect, and the resulting signals are combined with custom recorded background noise at various SNR levels. The model achieves an average accuracy of 94% with SNR $\in [-40, +10]$ dB, but the generalization performance to unseen non-synthetic data is not thoroughly assessed. A small-scale CNN architecture inspired by the VGG network (Simonyan and Zisserman, 2014) is proposed in Cantarini et al. (2021) to identify the Italian ambulance siren (belonging to the two-tone siren type) in noisy urban scenes with SNR $\in [-15, 0]$ dB. The model takes as input the STFT of a single channel audio signal, and reaches an average accuracy of 96.72% when trained and evaluated on synthetic data. To create the dataset, a single audio clip of the Italian siren is fed to a simulator that emulates acoustic propagation and Doppler effect, and recorded background noise is added afterwards. The authors also show that a classification accuracy of 86.87% is achieved when the model is exposed to SNR $\in [-30, -20]$ dB, unseen during training. However, the generalization ability to unseen siren types and real-world datasets is not evaluated.

Even though these systems exploit synthetic data for training siren identification networks, the effectiveness of their use is not thoroughly assessed by evaluating the model performance on unseen, purely real-world data. A similar assessment is carried out in Cantarini et al. (2022), where a model pre-trained on synthetic data is fine-tuned in a few-shot setting to recognize the Italian two-tone ambulance siren: although this scenario represents a realistic use-case of synthetic data and shows their effectiveness in the real-world application, it is tailored to a specific target sound.

2.2 Data augmentation strategies for audio classification

The use of synthetic data to train deep learning models is a form of data augmentation that consists of artificially crafting samples, either starting from existing datasets, or by generating data from a signal model. Several data augmentation techniques have been developed both in the computer vision (Shorten and Khoshgoftaar, 2019; Man and Chahl, 2022), and audio signal processing (Wei et al., 2020) fields. Due to the different nature of image and audio signals, domain-specific augmentation strategies have been developed, notwithstanding the existence of some common methods (Wei et al., 2020). In audio applications (e.g., audio classification, speech recognition, audio signal denoising), the most common techniques involve the application of (non-)linear transformation to either the raw audio or some time-frequency representation (e.g., spectrogram) of a recorded sample. Two types of transformations can be identified: the first category includes operations on a single audio segment, like time-stretching and pitch-shifting (Wei et al., 2020), or masking operations applied to the spectrogram (Park et al., 2019). The second category is based on combining multiple signals to obtain new audio samples. Adding white (or colored) noise to an audio signal, summing or temporally juxtaposing multiple audio samples to create a synthetic mixture, and interpolating between two or more audio segments are the main techniques that belong to this category (Wei et al., 2020).

A different approach to audio data augmentation is the synthetic generation of (spatial) acoustic scenes. This approach, usually adopted in indoor scenarios, is based on placing sound sources in a virtual acoustic environment (i.e., a room) and simulating the sound received by a listener located in an arbitrary point of the acoustic scene. For this purpose, room impulse responses between the position of the source and listener are simulated (or, alternatively and in presence of real rooms, recorded) and applied to recorded, anechoic audio samples representing the signals emitted by the sound sources to create synthetic scenes. (Koyama et al., 2022). Although this method is widely adopted in room acoustics, its use in outdoor spaces is limited due the challenges posed by the accurate physical simulation of moving sources, Doppler effect and realistic sounding urban environments (Damiano and van Waterschoot, 2022; Yin et al., 2023).

Within this paper we investigate synthetic data generation techniques for training siren identification models that generalize well to multiple real-world datasets. We propose two different data generation methods: the first one is based on the synthetic generation of (stationary) siren source signals, followed by the simulation of acoustic propagation, ground reflection and Doppler effect to emulate the behavior of a moving emergency vehicle. The second one, similar to (Cantarini et al., 2021; Marchegiani and Newman, 2022), relies on the use of a small set of recorded stationary siren sounds, collected in public databases, followed by the simulation of the above mentioned acoustic propagation effects. Finally, to craft realistic acoustic scenes, we superimpose to the simulated siren real-world urban background noise taken from the SONYC dataset (Cartwright et al., 2020) and evaluate noise augmentation strategies. In the next section, we introduce the details of the proposed data generation techniques and the CNN architectures.

3 Proposed methodology

We hereby introduce two distinct data generation strategies, that will be compared and evaluated in Section 4:

1. The first one consists in defining a parametric model for the generation of synthetic stationary siren source signals, where the term source signal refers to the emitted siren sound prior to any propagation effect; the motion of the emergency vehicle is then simulated by feeding these signals to the pyroadacoustics simulator, that provides an accurate emulation of acoustic propagation, ground reflection, air absorption and Doppler effect. This method allows to generate siren sounds without requiring any real-world recording, and the parametric model allows to create infinitely many different source signals.
2. The second one relies on the use of a small set of recorded siren sounds, provided as input to pyroadacoustics to emulate source motion. The recorded signals employed in this case should be stationary and clean from background noise. Whereas the source signal diversity is reduced compared to the first method, the recorded sirens provide a higher realism than those generated synthetically.

The only difference between the two methods lies therefore in the source signals used as input to the propagation simulator. In the next subsections, we describe the two components of the synthetic data generation, namely, the generation of the siren source signals, and the definition of the acoustic scene including the simulated moving source and the underlying background noise.

3.1 Synthetic siren signal generation

The siren sound is by its nature an artificial signal produced by means of electromechanical components. In particular, it is a harmonic signal composed of a fundamental frequency, modulated over time with a certain periodicity and modulation function that depends on the specific type of siren, and of a set of higher harmonic components. Given a generic siren signal, its discrete time-dependent fundamental frequency f_0 is controlled by the modulation function

$$f_0[k] = g[k], \tag{1}$$

where k denotes the discrete time index. Higher harmonics will be modulated similarly, and the $g[k]$ function will be specified below depending on the type of siren. We simulate all the three types of sirens, including harmonic components up to the Nyquist frequency (i.e., half of the chosen sampling frequency $f_s = 16\,\text{kHz}$), using a parametric model of each siren type.

1. The two-tone siren has a fundamental frequency that jumps between two constant values every half period (although more complex duty cycles exist, they are not considered in the current model) as

$$g[k] = f_{\text{low}} + \alpha f_{\text{low}} u[k - T_s f_s/2] \quad 0 \le k < T_s f_s, \tag{2}$$

where T_s denotes the period (in seconds) and $u[k]$ is the unit step function. The lower frequency f_{low} of the two-tones, the jump parameter α and the period duration T_s can vary within intervals specified by regional regulations. To create diversity within the class, we use these variables as free parameters: for each generated siren sample, we randomly draw a lower fundamental frequency f_{low} from a uniform distribution defined between 360 Hz and 900 Hz, set $\alpha = 1/3$ and randomly draw $T_s$from a uniform distribution defined on the interval $[0.5, 2]$s.

2. The wail siren has a fundamental frequency that continuously varies between two limit values f_{low} and f_{high}. The signal period is divided in two parts: a rise time T_{rise}, during which the

frequency is increased from f_{low} to f_{high}, and a fall time T_{fall} during which the opposite behavior is observed. Each half period can be simulated using a chirp signal with arbitrary type (linear, exponential, quadratic, or hyperbolic) that depends on the specific siren to be emulated. In particular, for the rising part with period T_{rise}, the function $g[k]$ takes the form

$$g[k] = \begin{cases} f_{\text{low}} + \left(f_{\text{high}} - f_{\text{low}}\right)\dfrac{k}{T_{\text{rise}} f_s} & \text{linear} \\ f_{\text{low}} + \left(f_{\text{high}} - f_{\text{low}}\right)\left(\dfrac{k}{T_{\text{rise}} f_s}\right)^2 & \text{quadratic} \\ f_{\text{low}} + \left(\dfrac{f_{\text{high}}}{f_{\text{low}}}\right)^{\frac{k}{T_{\text{rise}} f_s}} & \text{exponential} \\ \dfrac{f_{\text{high}} f_{\text{low}} T_{\text{rise}} f_s}{\left(f_{\text{low}} - f_{\text{high}}\right)k + f_{\text{high}} T_{\text{rise}} f_s} & \text{hyperbolic.} \end{cases} \tag{3}$$

For the falling part, we can derive similar equations by changing T_{rise} with T_{fall}, and inverting all occurrences of f_{low} and f_{high} in (3). The two parameters T_{rise} and T_{fall}, the frequency values f_{low}, f_{high}, and the chirp type are used as free parameters in the simulations. In particular, for each simulated sample we randomly draw $f_{\text{low}} \in [400, 800]$Hz, $f_{\text{high}} \in [1000, 2000]$Hz, $T_{\text{rise}} \in [0.1, 2.5]$s, $T_{\text{fall}} \in [0.7, 7]$s and a random chirp type.

3. The yelp siren behaves similarly to the wail but has a shorter period (thus resulting in a faster alternation between the low and high frequency); it can thus be simulated using the same model described in (3). We draw the signal parameters as follows: $f_{\text{low}} \in [400, 800]$Hz, $f_{\text{high}} \in [1000, 2600]$Hz, $T_{\text{rise}} \in [0.01, 0.15]$s, $T_{\text{fall}} \in [0.01, 0.15]$s, and use random chirp type.

The probability distributions from which the parameters are selected are uniform distributions on the above specified intervals. Moreover, all the mentioned parameter ranges have been picked by manual inspection of recorded real-world sirens, and randomness is used to maximize diversity and thus foster generalization.

3.2 Synthetic acoustic scene definition

Once the source signals have been either collected or generated, we create a synthetic acoustic scene resembling a real traffic environment. For the siren identification task the two relevant classes are the *noise* class, including all the possible sound events that contribute to the overall urban soundscape, except for sirens, and the *siren* class, including siren signals emitted by moving emergency vehicles, on top of the background noise. We create synthetic acoustic scenes as follows: we generate moving siren signals using the open-source[1] pyroadacoustics simulator (Damiano and van Waterschoot, 2022). This tool enables to simulate sound sources moving along arbitrary trajectories, together with the Doppler effect, the sound reflection produced by the road surface and the atmospheric sound absorption. Using either the recorded or the synthetic source signals described in Section 3.1 as input to the simulator, we generate 2 s-long audio samples that emulate moving sirens. To this end, in pyroadacoustics we consider an omnidirectional microphone and define a coordinate system centered in its position. Even though the microphone is stationary, the *relative* motion between source and microphone is simulated, thus the presence of moving microphones can be emulated (apart from the effect of wind noise caused by the air hitting the moving microphone) by using proper trajectories between the source and the receiver. For each simulated sample we define a random, smooth trajectory within a radius of 100 m from the position of the microphone, and choose a random speed between 0 m/s and 40 m/s. For these simulations we use either rectilinear trajectories, or quadratic Bézier curves.

Using this procedure and the two source signal types (synthetic or recorded) detailed above, we create two synthetic siren datasets. For the dataset based on the use of synthetic source signals (named SynSIR in the following), we generate 24 k siren samples, each generated using a different siren configuration (i.e., drawing different signal model parameters). For the one based on recorded source signals (named RecSIR), we collect a total of 47 clean, stationary siren clips (including 11 two-tone, 23 wail and 13 yelp clips) from www.freesound.org, and use them as input to pyroadacoustics. By drawing different random trajectories and speeds, we generate 24 k siren samples: in this case, even though the diversity of the source signals is significantly more limited than in the SynSIR dataset, the samples differ in the source trajectory and speed. We can thus interpret the simulation of acoustic propagation as a data augmentation tool.

For the environmental noise, we rely on the SONYC dataset (Cartwright et al., 2020), a large-scale collection of urban noise recorded in different locations in New York City. The recorded audio is provided in 10 s-long segments with accompanying labels that identify the audio events appearing in each recording, without temporal indications. We prune the dataset to exclude siren and alarm sounds, and collect a total of 50 h of noise data.

To create synthetic acoustic scenes using these (synthetic) siren and (recorded) noise datasets, we design a data loader, also employed to feed data to the models when training and apply noise augmentation. This component operates as follows.

- For each sample (of both siren and noise class), we randomly extract a 2 s-long noise background from the collected SONYC noise dataset.
- When performing noise augmentation, we draw a random number n_{th} from a uniform distribution defined on the interval [0,1] and use it as a threshold parameter: each time a sample is produced by the data loader, we apply the following noise augmentation procedure if $n_{\text{th}} \geq 0.6$. First, we extract a 2 s-long speech segment from the collected LibriSpeech dataset (Panayotov et al., 2015) and sum it to the background noise with a random amplitude in the range $[0, 1]$ (with uniform distribution). We then randomly decide whether to further augment it by simulating motion along a random trajectory using pyroadacoustics. The motion simulation is performed with probability 0.5. Similarly, we extract a random environmental noise event from the filtered

[1] https://github.com/steDamiano/pyroadacoustics/

TABLE 1 The modified VGGSir architecture, inspired by Cantarini et al. (2021). The number m of filters is set to 16 in the first layer; n_c denotes the number of target classes.

Layer	Number of blocks	Filters/ neurons
Conv2D 3 × 3, stride 1 + BN	2	m
MaxPool 2 × 2	-	-
Conv2D 3 × 3, stride 1 + BN	2	$2m$
MaxPool 2 × 2	-	-
Conv2D 3 × 3, stride 1 + BN	2	$4m$
MaxPool 2 × 2	-	-
GlobAvgPool	-	-
Fully Connected + BN	-	10
Output (FC)	-	n_c

TABLE 2 The modified BCResNet architecture, originally proposed in Kim et al. (2021) for acoustic scene classification. The number m is set to 30; n_c denotes the number of target classes.

Operator	Number of blocks	Filters/neurons
Conv2D 5 × 5, stride 2	-	$2m$
BC-ResBlock	2	m
MaxPool 2 × 2	-	-
BC-ResBlock	2	$1.5m$
MaxPool 2 × 2	-	-
BC-ResBlock	2	$2m$
BC-ResBlock	3	$2.5m$
Conv2D 1 × 1	-	n_c
GlobAvgPool	-	-

UrbanSound8K dataset (Salamon et al., 2014), randomly augment it using pyroadacoustics (again, with probability 0.5), and sum it to the background noise mixture with random amplitude in the range $[0, 1]$ (with uniform distribution) and random onset.

- For the siren class, we pick one sample from the desired siren dataset (either SynSIR or RecSIR, depending on the experiment), and add it to the background noise with a random SNR drawn from an uniform distribution defined on the interval $[-20, 0]$dB.

This procedure is performed online each time a sample is provided as input to the model during the training stage, in order to maximize the diversity of the samples seen by the network. We manually set the size of the thus generated dataset to 12 k samples: preliminary tests have shown that increasing the number of training samples does not benefit the performance of the models.

3.3 Siren identification architectures

To design an end-to-end siren identification system, we propose to introduce modifications to established audio classification CNN models in order to adapt them to our specific use-case, train them using the proposed data generation strategy and evaluate their performance on real datasets unseen during training. All three models take as input log-mel spectrogram features extracted from a 2 s-long audio segment (feature extraction will be discussed in Section 4.2). The first architecture (that we name VGGSir), inspired by the small-scale siren identification model introduced in Cantarini et al. (2021), is depicted in Table 1. The CNN architecture consists of three blocks, each containing two convolutional layers (Conv2D), followed by a max-pooling (MaxPool) operation. The number of convolutional filters is set to $m = 16$ in the layers of the first block and is doubled after each block. After the third block, a global average pooling (GlobAvgPool) (Lin et al., 2014) layer is used as an interface between the convolutional part and a fully-connected (FC) layer with 10 neurons, followed by an output layer with n_c neurons, one per target class. In addition to extending Cantarini et al. (2021) to target three siren types, we introduce some further optimizations. First, the GlobAvgPool layer replaces the Flatten operation used in Cantarini et al. (2021) and is adopted to prevent overfitting via the reduction of the number of model parameters: this constitutes a double advantage since it improves the network generalization ability while simultaneously shrinking the model size. Second, we introduce batch normalization (BatchNorm) layers after all the Conv2D and the first FC layer: this operation enhances the network convergence stability by re-scaling and re-centering the features after each layer. Finally, to prevent overfitting, we use dropout layers with drop probability 0.1 after each Conv2D layer, and with drop probability 0.5 after the first FC layer. The size of the resulting model is 72 954 parameters.

The second model is the BCResNet model, originally proposed in Kim et al. (2021) for the task of low-complexity acoustic scene classification in the DCASE 2021 challenge, Task 1 A (Martín-Morató et al., 2021), and is detailed in Table 2. This network relies on both 2D convolutions over the spectrogram features, and 1D convolutions over frequency-averaged embedded features. The processed 1D features are combined with the 2D ones by means of broadcasting operations and residual connections contained in the BC-ResBlock element (Kim et al., 2021). In our configuration, we introduce a dropout with drop probability 0.2 after each BC-ResBlock, and use $m = 30$ channels.

The third model is the BCResNorm model, a variation of the BCResNet originally introduced in Kim et al. (2021) for the same task. The difference with respect to BCResNet is the introduction of residual normalization operations for the input features and after each BC-ResBlock. The residual normalization is defined as follows. Given an input tensor $\mathbf{x} \in \mathbb{R}^{N\times C\times F\times T}$, where N, C, F, T represent the batch size, the number of channels, the number of frequency bins and time frames respectively, the instance normalization by frequency is defined as

$$\mathrm{FreqIN}\left(\mathbf{x}_{n,c,f,t}\right) = \frac{\mathbf{x}_{n,c,f,t} - \boldsymbol{\mu}_{n,f}}{\sqrt{\boldsymbol{\sigma}^2_{n,f} + \varepsilon}}, \tag{4}$$

where $\mathbf{x}_{n,c,f,t}$ denotes the element of $\mathbf{x}$ at position (n, c, f, t). We define $\boldsymbol{\mu}, \boldsymbol{\sigma}^2 \in \mathbb{R}^{N\times F}$ as the frequency-wise mean and standard deviation of the input feature $\mathbf{x}$, whose element at position (n, f) with $1 \leq n \leq N$ and $1 \leq f \leq F$ is defined, respectively, as

$$\boldsymbol{\mu}_{n,f} = \frac{1}{CT}\sum_{c=1}^{C}\sum_{t=1}^{T}\mathbf{x}_{n,c,f,t}, \tag{5}$$

$$\boldsymbol{\sigma}^2_{n,f} = \frac{1}{CT}\sum_{c=1}^{C}\sum_{t=1}^{T}\left(\mathbf{x}_{n,c,f,t} - \boldsymbol{\mu}_{n,f}\right)^2; \tag{6}$$

moreover, ε is a small constant introduced to avoid numerical instability. The idea of instance normalization by frequency is that in audio signals the domain differences are prominent along the frequency dimension (Kim et al., 2021), whereas in image processing they are captured by the *channel* mean and variance. The residual normalization operation is finally defined as

$$\text{ResNorm}\left(\mathbf{x}_{n,c,f,t}\right) = \lambda \cdot \mathbf{x}_{n,c,f,t} + \text{FreqIN}\left(\mathbf{x}_{n,c,f,t}\right) \tag{7}$$

where the weighting parameter λ is set to 0.1 as in Kim et al. (2021). The BCResNet and BCResNorm model have specifically been designed to target domain generalization in a classification problem. The choice of adopting and evaluating them for the siren identification task is supported by the fact that, when aiming at recognizing sirens in real-world data using models trained on synthetic data, we face a similar domain generalization problem. Similarly, the cross-dataset training and evaluation setting involves the generalization to a domain unseen during training. Both the BCResNet and BCResNorm architectures have 45 949 parameters and to the best of the authors' knowledge have never been adopted for the siren identification task.

4 Experimental validation

4.1 Data description

To evaluate the performance of the architectures with real-world data, we use three datasets specifically created for siren identification.

- The dataset for emergency siren classification presented in Asif et al. (2022) (we will refer to this dataset as LSSiren in the following), that contains 1800 files with duration ranging from 3 s to 15 s equally divided into *siren* and *noise* clips. The audio is either recorded in the wild, in a controlled environment or retrieved from online sources, and sirens of all the three types (two-tone, wail and yelp) are included (although only the binary labeling siren/noise is used).
- The sireNNet dataset presented in Shah and Singh (2023) and built for the classification of emergency vehicles. It contains 1,675 3 s-long files, 421 belonging to the noise class and the remaining ones to three different types of emergency vehicles (ambulance, fire-truck and police); however, the different vehicle types do not correspond to the three types of sirens (i.e., vehicles of the same type can present different siren patterns). We use the police and ambulance samples for the *siren* class: the fire-truck files are not considered for the experimental validation as they involve a massive presence of honking sounds, that have not been explicitly included in the data generation procedure and whose analysis is left for future work. Each siren sample in the sireNNet dataset is presented in two variants, one corresponding to the original recording, and the other one artificially augmented. To prevent the augmentation patterns from being learned by the networks, we use only the non-augmented files. Therefore, we use 847 files of this dataset (421 noise samples, and 426 siren samples) in total.
- An internal dataset recorded by Bosch (we will refer to this dataset as BoschSiren in the following), consisting of 2,970 3s-long siren files (including two-tone, wail and yelp sirens) and 400 noise files. The data has been recorded both in the wild and in a controlled environment in Germany, using a single microphone. Each file has been manually labeled as either *siren* or *noise.*

All files have been resampled to 16 kHz and converted to mono. Since we use 2 s-long audio segments as input to the CNNs, we apply the following pre-processing operations: for the LSSiren dataset we create 2 s splits of each file using a sliding window with length 2 s and no overlap; for the remaining two datasets we extract, from each file, a 2 s segment with a random onset. All datasets have been divided into train, validation and test data with ratios $[0.8, 0.1, 0.1]$: for the LSSiren dataset, the split is performed on the 1800 full-length files, and all the 2 s segments extracted from the same file have been assigned to the same split. In this manner we avoid assigning segments of the same file, that may have a high correlation, to different splits (this might in fact affect the performance evaluation).

4.2 Implementation details

To evaluate the different CNN architectures and training strategies, we implement the models described in Section 3.3 using the Pytorch (Paszke et al., 2019) framework. We choose the log-mel spectrogram as input feature (Figure 1): to compute it, we use torchaudio (Yang et al., 2021) and set a sampling frequency $f_s = 16\,\text{kHz}$, a window length of 0.064 s, a hop size of 0.032 s, 128 mel channels with $f_{\text{min}} = 300\,\text{Hz}$ and $f_{\text{max}} = f_s/2$. The choice of f_{min} is justified by the fundamental frequency of all sirens types being higher than 300 Hz: cutting the lower frequencies is thus exploited as a noise reduction technique. Furthermore, we normalize the input features using peak normalization to constrain the amplitude of the input features between 0 and 1, and we apply feature masking using SpecAugment (Park et al., 2019). In particular, we use two time masks and three frequency masks with maximum width of 10 consecutive frames for both dimensions.

When training models using synthetic data, we use four output units in all architectures ($n_c = 4$), corresponding to the noise, two-tone, wail and yelp siren classes. Due to the prominent difference among the time-frequency patterns of the three siren types, preliminary tests have shown that training the network to solve this multiclass classification problem, and thus to explicitly learn the differences among different siren types, leads to a higher accuracy also when evaluating the model on the binary classification problem (siren vs. noise), after training. In this case, the outputs of the two-tone, wail and yelp classes are merged into the single *siren* super-class. Since in the three real-world datasets only binary labels are

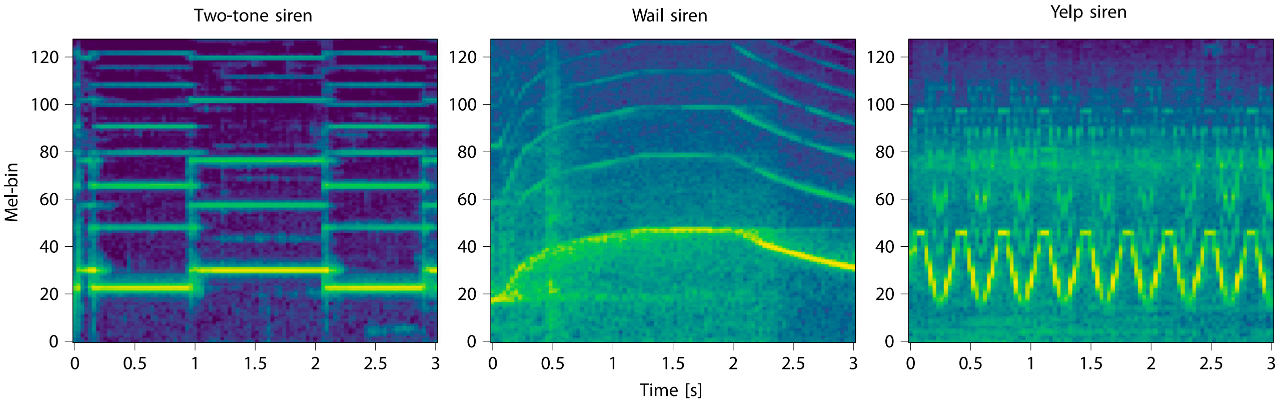

FIGURE 1
Log-mel spectrograms of three siren instances extracted from the LSSiren data. The log-mel spectrogram is adopted as input feature for the CNN models.

available, it was not possible to evaluate the models in the multiclass scenario, that is therefore used only when training on synthetic data. Therefore, all the results shown throughout this section will refer to the binary classification task.

We train all models for 15 epochs using the Adam optimizer (Kingma and Ba, 2015), the cross-entropy loss, and a batch size of 32. For each training process involving synthetic data, we split the training data into a training and validation set with ratio $[0.8, 0.2]$ and select the best model based on the loss on the validation set. We choose accuracy as evaluation metric in all the experiments (i.e., ratio between the correctly classified samples and the total number of samples).

4.3 Experimental evaluation

To set a reference for the evaluation of the training procedures employing synthetic data, as a first experiment we evaluate the performance of the three architectures in a cross-dataset setting. For this purpose, we only consider the three real-world datasets (BoschSiren, LSSiren, sireNNet): we train each model using one of them, and evaluate the performance on the other two. The results are reported in Figure 2, where we show, for each model and each training set, the accuracy range (min-max) on the two unseen test datasets, together with the average accuracy. Since the models are evaluated on data extracted from an unseen domain, which may have an underlying distribution different from that of the training set, the accuracy is degraded compared to models trained and evaluated on data extracted from the same dataset. This is illustrated in Table 3, where we compare the average accuracy obtained by the three architectures in the cross-dataset setting with the accuracy obtained on the test split of the same dataset used for training. We remark that, in this case, the train and test splits, though extracted from the same dataset, are disjoint. The results obtained on in-domain data (i.e., from the test split of the same dataset used for training) are comparable to those reported in the siren identification literature, where a similar evaluation setup is generally used. In the second column of this table we report the results averaged over all possible cross-dataset combinations (in total they amount to three, since we use one of the three datasets for training, and the other two for evaluation). From these two experiments we observe that the BCResNet model and its normalized variation achieve a higher accuracy compared to VGGSir, in accordance to the fact that these architectures are specifically designed to maximize the domain generalization performance. These architectures are thus promising for the task of siren identification. From Figure 2 we also observe that the BCResNorm model is characterized by narrower accuracy ranges, proving its higher robustness to data distribution shifts.

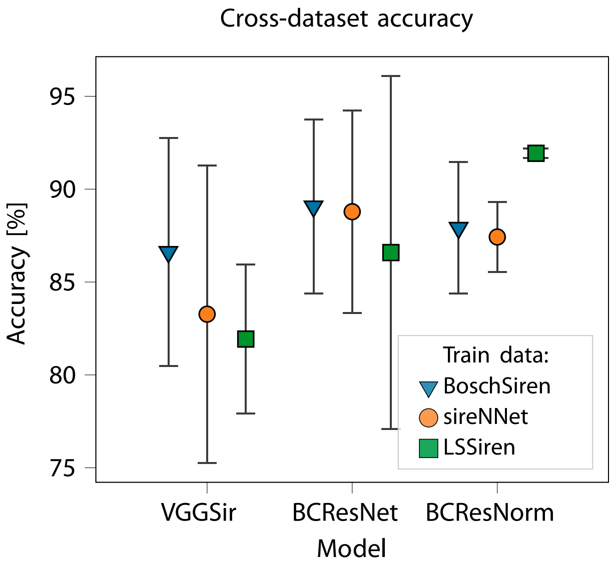

FIGURE 2
Baseline accuracy obtained with the three architectures (VGGSir, BCResNet, BCResNorm) evaluated in a cross-dataset setting: the models are trained using a single dataset and the accuracy score is computed on the other two. The markers show the average accuracy on the two test sets unseen during training, the whiskers denote the min-max interval.

TABLE 3 Comparison of the accuracy obtained when evaluating the models on in-domain data (i.e., using the test split of the same dataset used for training) and in a cross-domain setting (i.e., using the two datasets unseen during training). In the cross-domain evaluation, the results are averaged over all dataset combinations and the performance is degraded.

Model	In-domain accuracy [%]	Cross-domain accuracy [%]
VGGSir	98.01	83.10
BCResNet	98.08	88.14
BCResNorm	97.43	89.09

We then run an extensive evaluation campaign to assess the proposed training procedures based on the use of synthetic data. We thus train each model using either the SynSIR or the RecSIR datasets, and jointly evaluate the impact of applying noise augmentation. At training time, the best model is selected based on the minimum loss computed on an independent validation set generated from the same dataset used for training. In Table 4 we report, for each trained model, the accuracy obtained on the three real-world datasets, together with the average accuracy. First, we observe that training using the RecSIR data leads to a higher performance for all models: using recorded source signals represents therefore a better solution than generating synthetically, notwithstanding the limited diversity of the recorded sirens. This might be explained by the fact that the synthetic siren generation produces some artifacts that could be learned by the network, hindering its ability to generalize to real-world data, where these artifacts are not present. Second, we observe that the BCResNet and BCResNorm models achieve a higher performance than VGGSir, in line with the results of the previous experiment. Generalizing from synthetic to real-world data is in fact a different form of the same task of domain generalization tackled in the cross-dataset setting. BCResNorm shows also, once more, a narrower min-max accuracy range. Lastly, we observe that the impact of the noise augmentation strategy depends on the model and training data, and should thus be evaluated on a case-by-case basis when designing the training setup. The best-performing model is the BCResNet trained on RecSIR with noise augmentation, with an average accuracy of 93.73%: this constitutes a 4.64% improvement over the best model for the cross-dataset evaluation run in Table 3 (BCResNorm) and proves the effectiveness of the proposed training strategy. For each architecture, we select the best training configurations for training on SynSIR and RecSIR data from Table 4, based on the highest average accuracy. These models are compared with the cross-dataset training/evaluation setup in Figure 3. It can be observed that training using RecSIR always results in a higher accuracy compared to the cross-dataset training setup (+5.73% for VGGSir, +5.58% for BCResNet, +4.39% for BCResNorm), confirming the effectiveness of the proposed training strategy. For VGGSir and BCResNet, also the min-max range is reduced in this configuration. The proposed RecSIR data generation technique is therefore a preferable option for training siren identification models compared to the use of real-world datasets: with an extremely limited amount of training samples (47 publicly available audio clips) and an open-source sound propagation simulator, it is in fact possible to generate data that yield more robust and better performing models than using recorded datasets. On the other hand, training using SynSIR is not beneficial in terms of accuracy gain compared to the cross-dataset case. A study on how to improve synthetic source signal generation techniques is left for future work.

We finally evaluate the use of real-world data for the selection of the best model when training using synthetically generated data. For this experiment we re-train from scratch the BCResNet architecture using the RecSIR data with noise augmentation (i.e., the top-performing configuration from Table 4). This time, we select four different models by minimizing the validation loss computed on the RecSIR, BoschSiren, LSSiren, sireNNet validation sets, and evaluate them on the three real-world datasets. The results are reported in Figure 4. Surprisingly, selecting the model using data from the same dataset used for evaluation leads to the highest test accuracy only for the sireNNet dataset. For the LSSiren data, the best model is selected using the same RecSIR data, while for BoschSiren the four selection strategies lead to comparable performance. We also observe that selecting the model using the RecSIR data, though leading to the best choice only for the LSSiren test set, always produces a model with a performance close to that of the best choice: for BoschSiren data the accuracy drop is 1.04%, for sireNNet 1.88%, while for LSSiren the accuracy exhibits a gain of 0.49%. Using the RecSIR data constitutes an effective and robust choice, that keeps the training completely decoupled from the use for real-world datasets.

4.4 Performance investigation

In this section, we further investigate the performance of the best model selected in Table 4, namely, BCResNet trained using RecSIR data and noise augmentation (with RecSIR data used also for model selection). To this aim, we first compute, for each considered real-world dataset, the confusion matrix and evaluate the precision and recall scores. The precision metric is defined as

$$P = \frac{TP}{TP + FP}, \tag{8}$$

where TP represents the number of true positives and FP the number of false positives, whereas the recall is defined as

$$R = \frac{TP}{TP + FN}, \tag{9}$$

where FN denotes the number of false negatives. The precision and recall values for the three models are reported in Table 5. To establish a baseline for comparison, we compute the same

TABLE 4 Synthetic data evaluation procedure: the three analyzed architectures (VGGSir, BCResNet, BCResNorm) are trained using the synthetic datasets SynSIR and RecSIR. Augmentation of the background noise via the introduction of stationary and moving speech and environmental noise is evaluated for all combinations. The model selection is operated on validation data extracted from the same dataset used for training; the evaluation is performed on the three real-world datasets (LSSiren, BoschSiren and sireNNet), and the average accuracy is reported in the last column. The best models (i.e., the ones that yield the highest average accuracy) trained using RecSIR data are highlighted in bold; the best ones trained using SynSIR data are highlighted in italic.

Model	Siren signal	Noise augmentation	Accuracy LSSiren [%]	Accuracy BoschSiren [%]	Accuracy sireNNet [%]	Average accuracy [%]
VGGSir	SynSIR	-	87.17	52.86	76.88	72.30
	SynSIR	Yes	88.32	63.54	79.06	*76.97*
	RecSIR	-	87.83	87.76	90.94	**88.84**
	RecSIR	Yes	88.32	83.33	93.44	88.36
BCResNet	SynSIR	-	80.43	77.08	59.06	*72.19*
	SynSIR	Yes	82.73	54.69	65.94	67.79
	RecSIR	-	91.28	88.80	83.13	87.74
	RecSIR	Yes	96.71	91.67	92.81	**93.73**
BCResNorm	SynSIR	-	88.65	88.02	73.44	*83.37*
	SynSIR	Yes	88.65	86.72	72.19	82.52
	RecSIR	-	93.75	96.09	90.62	**93.49**
	RecSIR	Yes	91.94	95.57	83.44	90.32

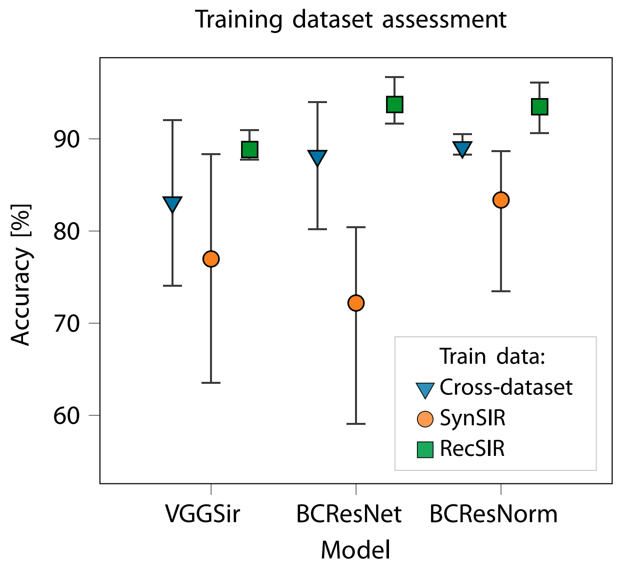

FIGURE 3
Performance comparison between moodels trained on the two synthetic datasets (SynSIR, RecSIR) and evaluated on the three real-world datasets: training on RecSIR always produces the highest average accuracy. The training dataset is specified in the legend, model selection is performed using validation data from the same dataset. The average accuracy on the real-world datasets is indicated by the markers, the whiskers denote the min-max interval. The average accuracy and min-max range obtained in a cross-dataset scenario are reported: in this case, the models are trained on one real-world dataset and evaluated on the remaining two.

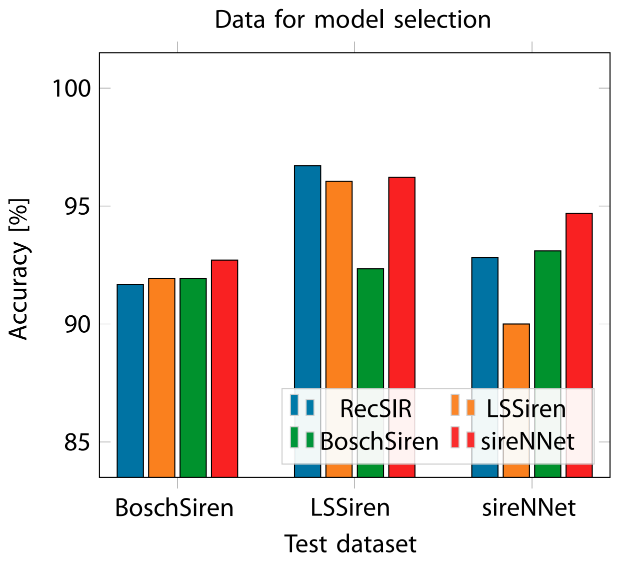

FIGURE 4
Performance of BCResNorm model trained on the RecSIR dataset and tested on the three real-world datasets (LSSiren, BoschSiren, sireNNet). The dataset used to select the best model during the training process is specified in the legend.

metrics using the same architecture trained on real-world data in a cross-domain setting. To this aim, for each test dataset we average the metrics achieved with the model trained on each of the other two datasets. Training with synthetic data always yields a higher precision than training with real data in a cross-dataset setting Although the recall score on the LSSiren and sireNNet datasets obtained using cross-dataset training is higher, the one on the BoschSiren data is strongly degraded. This might be caused by the sireNNet and BoschSiren data having a more similar underlying distribution among each other, and shifted from the BoschSiren one. The recall obtained on BoschSiren using synthetic training is instead

TABLE 5 Evaluation of precision and recall scores for the BCResNet model trained on RecSIR data with noise augmentation and evaluated on the three real-world datasets. Training on the RecSIR dataset leads to the best average scores, highlighted in bold.

Test data	RecSIR		Cross-dataset	
	Precision	Recall	Precision	Recall
LSSiren	0.9876	0.9336	0.9156	0.9453
BoschSiren	0.9922	0.8951	0.9731	0.7570
sireNNet	0.9926	0.8590	0.8311	0.9743
Average	**0.9908**	**0.8959**	0.9066	0.8922

comparable with the performance achieved on the other two datasets. This suggests that the synthetic training strategy is more robust to a data distribution shift, confirming its potential to train models that can generalize to unseen distributions.

We finally analyze the precision-recall (PR) curves obtained using the BCResNet model trained on RecSIR data with noise augmentation, plotted for the three datasets in Figure 5, where the area under the curve (AUC) is also reported. We highlight three operating points on the PR curve: the equal error rate point, where $R = P$, and the two points in which either the precision or the recall exceeds a threshold of 0.95. For the autonomous driving use-case, safety arguments suggest that a higher recall is more relevant than a higher precision: a missed detection is in fact more dangerous than a false positive. Furthermore, in a multimodal context in which information collected from more than one sensor is merged to compute the final prediction, a missed detection from one sensor may reduce the overall robustness of the system. To get a higher recall at the cost of a reduced precision (i.e., the risk of having more false positives) the operating point where $R \geq 0.95$ might therefore be preferred. On the other hand, if the system is used to alert human drivers, false positives may be less tolerable as they might distract the driver. Thus, a higher precision might be preferred, assuming that missed detection can still be caught by the driver. In this case, the operating point where $P \geq 0.95$ might be preferred.

5 Conclusion

In this work we have addressed the task of emergency siren identification for automotive applications using deep learning. Though the topic is well-known in the literature, state-of-the-art models are usually trained and evaluated on data extracted from the same dataset. Due to the prominent differences that can be observed among siren sounds and background noise mixtures in different locations around the world, collecting a dataset that accurately represents all the variations is in practice unfeasible. To reduce the burden of data collection, we hence proposed to train models using synthetic data: we introduced two synthetic data generation strategies, the first one based on the synthetic generation of stationary siren signals, and the second one based on the collection of a limited number of samples of stationary sirens from public repositories. In both procedures, sound propagation and Doppler effect are then emulated using a road acoustics simulator to create synthetic datasets that can be used to train siren identification models. To evaluate the two methods, we selected state-of-the-art CNNs for siren identification and acoustic domain generalization: we introduced several modifications to enhance the performance of these models on our target task, and trained them using synthetic datasets crafted by means of the two proposed data generation strategies. Finally, we showed that using recorded stationary sirens as source signals and simulating acoustic propagation to create an augmented dataset yields the best performance using all analyzed networks. In particular, training all models using synthetic data generated using stationary recorded sirens and synthetic acoustic propagation is effective: the accuracy obtained using this proposed training method is in fact higher than the one obtained when training the models using real-world data and evaluating them in a cross-dataset setting. As an additional advantage, training using synthetic data cuts the costs of data collection. The combination of the

The proposed training strategy paves the way for future research: first, the proposed method can be extended to target the identification of additional alarm signals (e.g., car horns) in urban environments. Furthermore, the trade-off between the

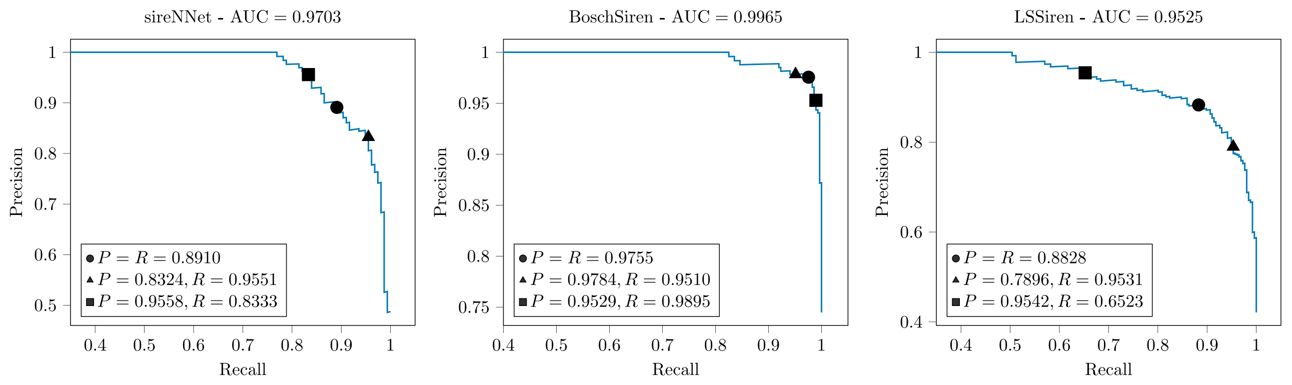

FIGURE 5
Precision-recall (PR) curves obtained with the BCResNet model trained on RecSIR data with noise augmentation and evaluated on the three real-world datasets. Three operating points, where $P = R$, $P \geq 0.95$ or $R \geq 0.95$, are highlighted with markers. The AUC is also reported for the three datasets.

model complexity and performance has not been investigated: the synthetic data generation comes with the advantage of enabling to generate datasets with arbitrary sizes and to define arbitrarily complex acoustic scenes. The design of more elaborate acoustic scenes, together with the generation of larger (and more diverse) datasets, might therefore foster the generalization performance of the analyzed models or, alternatively, promote the design of larger models, at the cost of a higher complexity. Finally, the use of generative models to improve the quality of the simulated data might boost the effectiveness of the methods discussed throughout this work.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://doi.org/10.6084/m9.figshare.19291472 https://data.mendeley.com/datasets/j4ydzzv4kb/1.

Author contributions

SD: Conceptualization, Data curation, Formal Analysis, Methodology, Software, Visualization, Writing–original draft. BC: Methodology, Writing–review and editing. AG: Resources, Supervision, Writing–review and editing. TW: Conceptualization, Funding acquisition, Methodology, Supervision, Writing–review and editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956962, from the European Union's Horizon Europe research and innovation programme under the grant agreement No. 101070374, and from the European Research Council under the European Union's Horizon 2020 research and innovation program/ERC Consolidator Grant: SONORA (no. 773268).

Acknowledgments

The authors wish to thank Dr. Luca Bondi (Bosch Research) for the useful discussions that helped in shaping this manuscript.

Conflict of interest

Authors BC and AG were employed by Robert Bosch GmbH.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author disclaimer

This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

Publisher's note

References

Asif, M., Usaid, M., Rashid, M., Rajab, T., Hussain, S., and Wasi, S. (2022). Large-scale audio dataset for emergency vehicle sirens and road noises. *Sci. Data* 9, 599. doi:10.1038/s41597-022-01727-2

Beritelli, F., Casale, S., Russo, A., and Serrano, S. (2006). "An automatic emergency signal recognition system for the hearing impaired," in Proc. 2006 IEEE 12th digital signal process. Workshop & 4th IEEE signal process. Education workshop *(teton national park, WY, USA)*, 179–182. doi:10.1109/DSPWS.2006.265438

Cantarini, M., Brocanelli, A., Gabrielli, L., and Squartini, S. (2021). "Acoustic features for deep learning-based models for emergency Siren detection: an evaluation study" in Proc. 2021 12th int. Symp. Image signal process. Anal. (ISPA), zagreb, Croatia, 47–53.

Cantarini, M., Gabrielli, L., and Squartini, S. (2022). Few-shot emergency Siren detection. *Sensors* 22, 4338. doi:10.3390/s22124338

Carmel, D., Yeshurun, A., and Moshe, Y. (2017). "Detection of alarm sounds in noisy environments," in *Proc. 25th European signal process. Conf. (EUSIPCO) (kos, Greece)*, 1839–1843. doi:10.23919/EUSIPCO.2017.8081527

Cartwright, M., Cramer, J., Mendez, A. E. M., Wang, Y., Wu, H.-H., Lostanlen, V., et al. (2020). *SONYC Urban Sound Tagging (SONYC-UST): a multilabel dataset from an urban acoustic sensor network*. doi:10.5281/zenodo.3966543

Damiano, S., and van Waterschoot, T. (2022). "Pyroadacoustics: a road acoustics simulator based on variable length delay lines," in *Proc. 25th int. Conf. Digital audio effects (DAFx20in22) (vienna, Austria)*, 216–223.

DIN14610:2022-03 (2022). *Sound warning devices for authorized emergency vehicles*. doi:10.31030/3325071

Fazenda, B., Atmoko, H., Gu, F., Guan, L., and Ball, A. (2009). "Acoustic based safety emergency vehicle detection for intelligent transport systems," in Proc. 2009 ICCAS-SICE *(fukuoka, Japan)*, 4250–4255.

Furletov, Y., Willert, V., and Adamy, J. (2021). "Auditory scene understanding for autonomous driving," in *Proc. 2021 IEEE intelligent vehicles symp. (IV) (nagoya, Japan)*, 697–702. doi:10.1109/IV48863.2021.9575964

Hussain, R., and Zeadally, S. (2019). Autonomous cars: research results, issues, and future challenges. *IEEE Commun. Surv. Tutorials* 21, 1275–1313. doi:10.1109/COMST.2018.2869360

Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). *Synthetic data and artificial neural networks for natural scene text recognition*. arXiv preprint arXiv:1406.2227. doi:10.48550/arXiv.1406.2227

Kim, B., Yang, S., Kim, J., and Chang, S. (2021). "Domain generalization on efficient acoustic scene classification using residual normalization," in *Proc. Detection classification acoustic scenes events (DCASE2021)*.

Kingma, D. P., and Ba, J. (2015). "Adam: a method for stochastic optimization," in *Proc. 3rd int. Conf. Learning representations (ICLR) (san diego, USA)*. doi:10.48550/arXiv.1412.6980

Koyama, Y., Shigemi, K., Takahashi, M., Shimada, K., Takahashi, N., Tsunoo, E., et al. (2022). "Spatial data augmentation with simulated room impulse responses for sound event localization and detection," in *Proc. 2022 int. Conf acoust., speech and signal process. (ICASSP)*, 8872–8876. doi:10.1109/ICASSP43922.2022.9746754

Lin, M., Chen, Q., and Yan, S. (2014). *Network in network*. arXiv preprint arXiv.1312.4400. doi:10.48550/arXiv.1312.4400

Man, K., and Chahl, J. (2022). A review of synthetic image data and its use in computer vision. *J. Imaging* 8, 310. doi:10.3390/jimaging8110310

Marchegiani, L., and Fafoutis, X. (2022). How well can driverless vehicles hear? A gentle introduction to auditory perception for autonomous and smart vehicles. *IEEE Intell. Transp. Syst. Mag.* 14, 92–105. doi:10.1109/MITS.2021.3049425

Marchegiani, L., and Newman, P. (2022). Listening for sirens: locating and classifying acoustic alarms in city scenes. *IEEE Trans. Intelligent Transp. Syst* 23, 17087–17096. doi:10.1109/TITS.2022.3158076

Martín-Morató, I., Heittola, T., Mesaros, A., and Virtanen, T. (2021). *Low-complexity acoustic scene classification for multi-device audio: analysis of DCASE 2021 challenge systems*. arXiv preprint arXiv.2105.13734. doi:10.48550/arXiv.2105.13734

Meucci, F., Pierucci, L., Del Re, E., Lastrucci, L., and Desii, P. (2008). "A real-time siren detector to improve safety of guide in traffic environment," in *Proc. 16th European signal process. Conf.* (Lausanne, Switzerland: EUSIPCO), 1–5.

Nandwana, M. K., and Hasan, T. (2016). "Towards smart-cars that can listen: abnormal acoustic event detection on the road," in *Proc. Interspeech 2016 (san francisco, USA)*, 2968–2971. doi:10.21437/Interspeech.2016-1366

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). "Librispeech: an asr corpus based on public domain audio books," in *Proc. 2015 IEEE int. Conf. Acoust. Speech signal process. (ICASSP) (south brisbane, QLD, Australia)*, 5206–5210. doi:10.1109/ICASSP.2015.7178964

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., et al. (2019). "SpecAugment: a simple data augmentation method for automatic speech recognition," in *Proc. Interspeech 2019 (graz, Austria)*, 2613–2617. doi:10.21437/Interspeech.2019-2680

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). "Pytorch: an imperative style, high-performance deep learning library," in *Proc. 33rd int. Conf. Neural information processing syst. (red hook, NY, USA)*.

Piczak, K. J. (2015). "Environmental sound classification with convolutional neural networks," in *Proc. 2015 IEEE 25th int. Workshop machine learning for signal process. (MLSP) (boston, MA, USA)*, 1–6. doi:10.1109/MLSP.2015.7324337

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *Proc. Medical image computing and computer-assisted intervention (MICAAI 2015) (munich, Germany)*, 234–241. doi:10.1007/978-3-319-24574-4_28

Salamon, J., Jacoby, C., and Bello, J. P. (2014). "A dataset and taxonomy for urban sound research," in *Proc. 22nd ACM int. Conf. Multimedia (ACM-MM'14) (orlando, FL, USA)*, 1041–1044. doi:10.1145/2647868.2655045

Schröder, J., Goetze, S., Grützmacher, V., and Anemüller, J. (2013). "Automatic acoustic siren detection in traffic noise by part-based models," in *Proc. 2013 IEEE int. Conf. Acoust. Speech signal process. (ICASSP) (vancouver, BC, Canada)*, 493–497. doi:10.1109/ICASSP.2013.6637696

Shah, A., and Singh, A. (2023). sireNNet-emergency vehicle Siren classification dataset for urban applications. doi:10.17632/j4ydzzv4kb.1

Sharma, J., Granmo, O.-C., and Goodwin, M. (2021). "Emergency detection with environment sound using deep convolutional neural networks," in *Proc. 5th int. Congr. Inf. Commun. Technol. (London, UK)*, 144–154. doi:10.1007/978-981-15-5859-7_14

Shorten, C., and Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *J. Big Data* 6, 60. doi:10.1186/s40537-019-0197-0

Simonyan, K., and Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556. doi:10.48550/arXiv.1409.1556

Sun, H., Liu, X., Xu, K., Miao, J., and Luo, Q. (2021). *Emergency vehicles audio detection and localization in autonomous driving*. arXiv preprint arXiv:2109.14797. doi:10.48550/arXiv.2109.14797

Tran, V.-T., and Tsai, W.-H. (2020). Acoustic-based emergency vehicle detection using convolutional neural networks. *IEEE Access* 8, 75702–75713. doi:10.1109/ACCESS.2020.2988986

Walden, F., Dasgupta, S., Rahman, M., and Islam, M. (2022). Improving the environmental perception of autonomous vehicles using deep learning-based audio classification. *arXiv Prepr. arXiv:2209.04075*. doi:10.48550/arXiv.2209.04075

Wei, S., Zou, S., Liao, F., and Lang, W. (2020). A comparison on data augmentation methods based on deep learning for audio classification. *J. Phys. Conf. Ser.* 1453, 012085. doi:10.1088/1742-6596/1453/1/012085

Yang, Y.-Y., Hira, M., Ni, Z., Chourdia, A., Astafurov, A., Chen, C., et al. (2021). Torchaudio: building blocks for audio and speech processing. *arXiv Prepr. arXiv:2110.15018*. doi:10.48550/arXiv.2110.15018

Yin, J., Damiano, S., Verhelst, M., van Waterschoot, T., and Guntoro, A. (2023). "Real-time acoustic perception for automotive applications," in *Proc. 2023 design, automation & test in Europe conf. & exhibition (DATE) (antwerp, Belgium)*, 1–6. doi:10.23919/DATE56975.2023.10137209