Deployment of AI-based Sound Analysis Algorithms in Real-time Acoustic Sensors: Challenges and a Use Case

Amaia Sagasti[1], *Pere Artís*[2], *Xavier Serra*[1], *Frederic Font*[3]

[1]Music Technology Group, Universitat Pompeu Fabra, Barcelona [2] Keacoustics, Barcelona
[3]Phonos Fundació Privada, Barcelona

***Abstract*—Real-time acoustic sensing involves significant challenges in capturing, processing, and transmitting audio. Integrating AI models on resource-constrained devices further complicates development. This paper presents an end-to-end solution addressing these challenges: SENS, the Smart Environmental Noise System, is a low-cost sensor designed for real-time acoustic monitoring. Built on a *Raspberry Pi* platform, SENS captures sound continuously and processes it locally using custom-developed software based on small and efficient artificial intelligence algorithms. With a current focus on urban environments, SENS calculates acoustic parameters, including sound pressure level (SPL), and makes predictions of the perceptual sound attributes of *pleasantness* and *eventfulness* (ISO 12913), along with detecting the presence of specific sound sources such as vehicles, birds, and human activity. To safeguard privacy, all processing occurs directly on the device in real-time ensuring that no audio recordings are permanently stored or transferred. Additionally, the system transmits the analysis results through the wireless network to a remote server. Demonstrating its practical applicability, a network of five SENS devices has been deployed in an urban area for over three months, validating SENS as a powerful tool for analyzing and understanding soundscapes, recognizing patterns, and detecting acoustic events. The proposed flexible and reproducible technology allows reconfiguration for different applications and represents an innovative step in real-time and AI-based noise monitoring.**

***Index Terms*—Environmental noise monitoring, machine learning, Internet of Things (IoT), urban soundscapes, smart city**

1. INTRODUCTION

Sound monitoring has traditionally relied on high-precision instruments such as sonometers, but these are limited by their high cost, lack of remote communication capabilities (requiring manual deployment and retrieval), and low spatial or temporal resolution. The growing adoption of Internet of Things (IoT) technologies has dramatically transformed this paradigm. Recent developments in low-power embedded systems and wireless communication have enabled distributed sensing via low-cost acoustic sensor networks. Examples like AudioMoth [1] have made long-term acoustic data collection feasible, but they often still rely on offline analysis.

The parallel emergence of artificial intelligence (AI) across numerous fields opens new opportunities for real-time and autonomous sound analysis. Approaches like Bonet-Solà et al. [2] integrate noise level data from public wireless sensor networks with short audio clips recorded on smartphones to estimate subjective acoustic comfort. However, their system relies on centralized processing and offline AI models. Similarly, the CENSE network [3], [4], uses MEMS-based sensors to transmit low-resolution spectral data, ensuring privacy while relying on centralized servers for analysis. Another example of this approach is the LIFEWARD project [5] for neonatal ICUs, which computes third-octave spectrograms on the edge to avoid storing intelligible audio, with cloud-based AI completing the analysis. In contrast, other solutions involve AI-powered sensors that can perform on-device sound event detection, noise classification, or perceptual indexing. An example of this is a system deployed around the VELTINS Arena in Germany [6], which runs lightweight convolutional neural networks (CNNs) on Raspberry Pi devices to classify sound events locally, reducing both data transmission and privacy risks. This shift enables real-time, autonomous, and privacy-preserving insights but introduces challenges related to computational efficiency.

In this work, we present a methodology for deploying AI-based sound analysis algorithms in real-time acoustic sensors through a use case: SENS (Smart Environmental Noise System), a low-cost acoustic sensor system designed to run lightweight AI models locally for continuous sound analysis. Built on Raspberry Pi, and with a current focus on urban environments, it estimates both physical (e.g., sound pressure level) and perceptual attributes (e.g., *pleasantness*, *eventfulness*), as well as the sound sources present in the soundscape, while preserving privacy by avoiding permanent storage or transmission of audio. Results are transmitted through the network to a remote server, and the system is modular to support flexible model updates towards other applications. SENS technology is validated through a real-world urban deployment, demonstrating its potential for scalable, real-time acoustic monitoring.

2. SYSTEM OVERVIEW

The fully integrated proposed technology involves both hardware and software components. However, the software does not require dedicated hardware and, depending on the specific application, can run on a standard laptop or any device with an audio input. Additionally, if transmitting data to a remote server is unnecessary, the modular design of the software allows it to operate entirely offline.

Each SENS sensor constitutes a low-cost solution built around a single *Raspberry Pi*. The device captures sound through a connected microphone and transmits results via a mobile network hat with a SIM card. Sensors can be accessed remotely through a virtual private network (VPN), allowing over-the-air software updates and problem resolution. Besides, a hardware *watchdog* allows for autonomous system reboots if certain conditions are not met (e.g., if there are connectivity issues). For the use case presented in this paper (see Section 6), the sensors were connected to a continuous power supply allowing uninterrupted operation, though solutions that make use of batteries were also developed. The Github repository of the project[1] contains guidelines for building custom SENS hardware devices.

The software is developed end-to-end with a modular design that allows flexibility: three independent but related processes run in parallel — sound capture, processing, and transmission of results to a remote server. The code is implemented in Python and is available in the *sens-sensor* GitHub repository[1]. The modular software structure in each SENS device is further explained in the following sections.

3. AUDIO ACQUISITION

The sound capture process is straightforward. Audio is continuously recorded and, when the audio buffer reaches a defined duration (3 seconds in the SENS implementation), it is saved as a *pickle* file in a

[1] https://github.com/MTG/sens-sensor

specified folder, with a filename that includes the date and time for reference. The equivalent continuous sound levels, L_{eq} and L_{Aeq}, are computed for each segment and saved in a file with the same naming convention in the same directory. While incoming audio frames are being saved to disk, a dedicated thread continuously monitors the saved audio files and deletes those older than a defined retention period (30 seconds in SENS). This approach improves data privacy by ensuring that the sensor does not permanently store audio data, while also improving the device's storage efficiency. Microphone calibration is essential, as it directly affects the calculation of L_{eq} and L_{Aeq}, and, as shown in an earlier study [7], it can influence model accuracy.

4. AUDIO PROCESSING

The processing module is a crucial part of the acoustic monitoring solution, as its development requires addressing several key challenges. To safeguard data privacy, all audio processing is performed locally and no audio is permanently stored or transmitted. Running machine learning models in real-time on a low-cost device with limited computational resources demands the development of lightweight models. Additionally, to achieve adaptability to different monitoring use cases, independent and separate models are trained for each task, resulting in a flexible modular architecture. The following subsections detail the research and training of the lightweight models, followed by their implementation within the software architecture.

4.1. Model training

The use case for which SENS has been developed is focused on the monitoring of urban spaces. Therefore, we developed sound analysis algorithms that predict perceptual soundscape attributes, *pleasantness* and *eventfulness*; and estimate the saliency of common sound sources present in the acoustic environment: *birds*, *construction works*, *dogs*, *human activity*, *sirens*, *music* and *vehicles*. For this purpose, we used existing datasets with open licenses. The ARAUS dataset [8] is used for training *pleasantness* and *eventfulness* models. It consists of 30s-length augmented, but realistic, soundscape audios, each labeled with values of pleasantness[-1, 1] and eventfulness[-1, 1], obtained following the soundscape study methodology suggested in ISO-12913 [9]–[11]. For the estimation of sound sources, we used several publicly available datasets. The Urban Sound Monitoring (USM) dataset [12] was used for *birds*, *construction works*, *dogs*, *human activity*, *sirens*, and *music*. This dataset consists of 5-second polyphonic stereo soundscapes composed of sounds from the FSD50k dataset [13]. Before training, we adapted the dataset by removing irrelevant sound sources, such as *gunshot*, and mapping more specific classes to general ones—for example, *cheering*, *scream*, and *speech* were all mapped to *human*. For each sound source, we built an independent model using a one-vs-all classification approach, allowing the sensor to be customized for different applications.

The approach to build the *vehicles* sound source model involved the careful combination of two datasets. A set of audios from the IDMT Traffic dataset [14] was selected, each consisting of 2-second long stereo audio recordings of vehicle sounds (*bus*, *car*, *motorcycle*, and *truck*). The balanced selection of IDMT audios was combined as an additional sound class within the UrbanSound8k (US8k) dataset [15]. This dataset originally includes sound excerpts ($<= 4s$) of urban sounds from 10 classes, including noisy sounds like *air conditioner*, *drilling* and *engine idling*, that were cut to 2 seconds long. Again, the vehicles model is trained following a one-vs-all approach.

Despite the various datasets used for each parameter, the remaining training process was consistent across all. The algorithms take as input sound representations generated using Laion-AI's CLAP (Contrastive Language-Audio Pretraining) *630k-fusion-best* model [16]. Previous research has demonstrated that this representation performs well in similar classification tasks [7], [17]. The CLAP model produces a 512-dimensional embedding vector, denoted as:

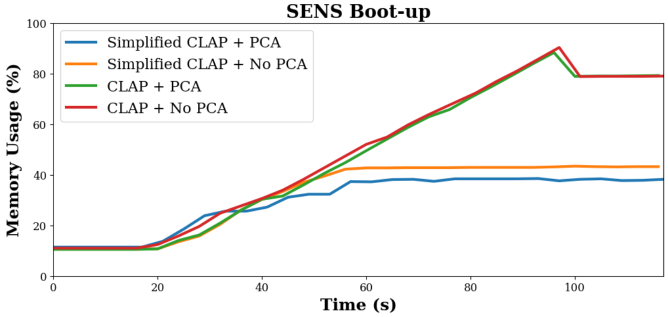


Fig. 1: When SENS boots up, all machine learning models (CLAP and individual, both perceptual and source detection, models) load before starting normal operation. The graph shows the device's memory usage for different configurations with original CLAP or with simplified CLAP model (which only includes the audio encoder), and the use, or not, of PCA for reducing embeddings' dimensionality.

$$E = \{E_0, E_1, \ldots, E_{511}\}, \quad E \in \mathbb{R}^{512} \tag{1}$$

Due to the SENS device's limited computational capability, using the original CLAP model along with its raw embeddings caused the memory load to reach its limit, frequently resulting in system freezes. CLAP models function by learning a joint embedding space for both audio and textual descriptions. The original LAION-AI's CLAP model consists of two branches: one for converting text to an embedding space and another for converting audio into the same embedding space. To optimize our use case, since bi-directional matching was not required, we removed the text encoder. By doing so, we significantly reduced the model's memory consumption, making it more feasible for real-time processing on the Raspberry Pi without compromising accuracy.

To further optimize memory usage, we used Principal Component Analysis (PCA) to reduce the dimensionality of the embedding space. A PCA transformation was derived from an analysis of over 25,000 audio samples from the ARAUS dataset. The results indicated that 50 principal components were sufficient to explain 95.46% of the data variability, with a negligible effect on the prediction accuracy. The reduced embedding vector is given by:

$$E' = \{E'_0, E'_1, \ldots, E'_{49}\}, \quad E' \in \mathbb{R}^{50} \tag{2}$$

Figure 1 illustrates the sensor's boot-up under four different configurations, reflecting the impact that cleaning the CLAP model and applying PCA have on the boot-up time and memory load.

Using our reduced embedding space, a variety of simple classifiers were evaluated for the different properties that SENS estimates, with final selections including Random Forest Regressor, Linear Support Vector Classification, and Logistic Regression with LBFGS optimization. *Pleasantness* and *eventfulness* were treated as regression problems, with outputs ranging [-1,1] where -1 corresponds to unpleasant and uneventful, and 1 represents pleasant and eventful, respectively. On the other hand, the sound sources approach followed a one-vs-all classification. Their output ranges within [0,1] representing the model's estimated likelihood that the input belongs to the positive class. The resulting models achieve Mean Absolute Errors (MAE) of

Table 1: Summary of datasets and regressors/classifiers used for training models to predict various parameters, with corresponding validation metrics. Algorithm acronyms: Random Forest Regressor (RFR), Support Vector Classification (SVC), Logistic Regression (LR).

Parameter	Dataset	Algorithm	Metrics (val)
Pleasantness	ARAUS	RFR	0.22 MAE
Eventfulness			0.20 MAE
Birds	USM	Linear SVC	97% precision
Construction			81% precision
Dogs			92% precision
Human			83% precision
Sirens			88% precision
Music		LR - LBFGS	80% precision
Vehicles	IDMT-Traffic, US8k	Linear SVC	100% precision

0.22 and 0.20 on the validation sets for *pleasantness* and *eventfulness*, respectively, and the sound sources classification models achieve precision scores (i.e., the proportion of correctly classified positive samples) that exceed 80% across all categories on the validation set (Table 1). The code used for training the models is available in the project's Github repository.

4.2. Model deployment

The implementation of the trained AI models is relatively straightforward. A dedicated thread continuously monitors the folder where audio files are saved by the capturing module, waiting for new incoming data. As soon as a new file appears, the audio data is read to generate a reduced embedding vector E'. This is passed to the set of AI models, each of which outputs a prediction value. Due to the fact that *pleasantness* and *eventfulness* constitute integrated perceptions of the soundscape rather than instantaneous measurements, the processing module also aggregates the 10 most recent audio frames (corresponding to 30 seconds of audio data in the SENS use case). Thus, another reduced embedding vector is generated and passed to the *pleasantness* and *eventfulness* models, obtaining predictions of these parameters over a longer period. Finally, integrating the previously saved sound levels, L_{eq} and L_{Aeq}, the module generates an output dictionary with all the compiled results. This is stored in a JSON file within a designated folder.

5. DATA TRANSMISSION

For data transmission, each SENS device in the use case network operates remotely and sends the analysis results to a remote server for storage and display on a web platform. This requires appropriate hardware and poses challenges, particularly in terms of internet data consumption. While HTTPS is widely used because it ensures secure data transfer, it can add significant overhead due to large headers (5–10KB). For example, each JSON file generated by the processing module for a 3-second audio chunk is about 650 bytes. Sending each result individually at a rate of 20 messages per minute would result in over 8GB of monthly data usage per sensor. To mitigate this, multiple JSON files are batched together into a single HTTPS request, significantly reducing data consumption to 2-3GB per month.

The data transmission module in the proposed solution consists of a script which continuously monitors the folder where analysis results are stored and sends them to the remote server once a number of files are accumulated (10 in our implementation). Their contents are combined into a single payload and transmitted to a remote server via the mobile network. If the network connection is unavailable, audio acquisition and processing continue locally, and transmission resumes automatically once the connection is restored. Upon successful receipt, the transmitted JSON files are deleted locally. The custom server stores all incoming data in a database and offers a visualization tool that enables users to monitor active sensors, check real-time status metrics like memory and CPU usage, and visualize processed data with interactive graphs. A detailed description of this server framework is beyond the scope of this paper.

6. REAL-WORLD DEPLOYMENT

A network of five SENS devices has been deployed in the real-world as part of *Smart Iruña Lab* [18], a smart cities program carried out in the city of Pamplona/Iruña. With over three months of deployment, SENS technology has proven to be a reliable tool for noise monitoring.

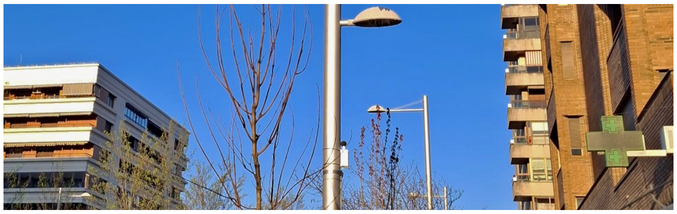
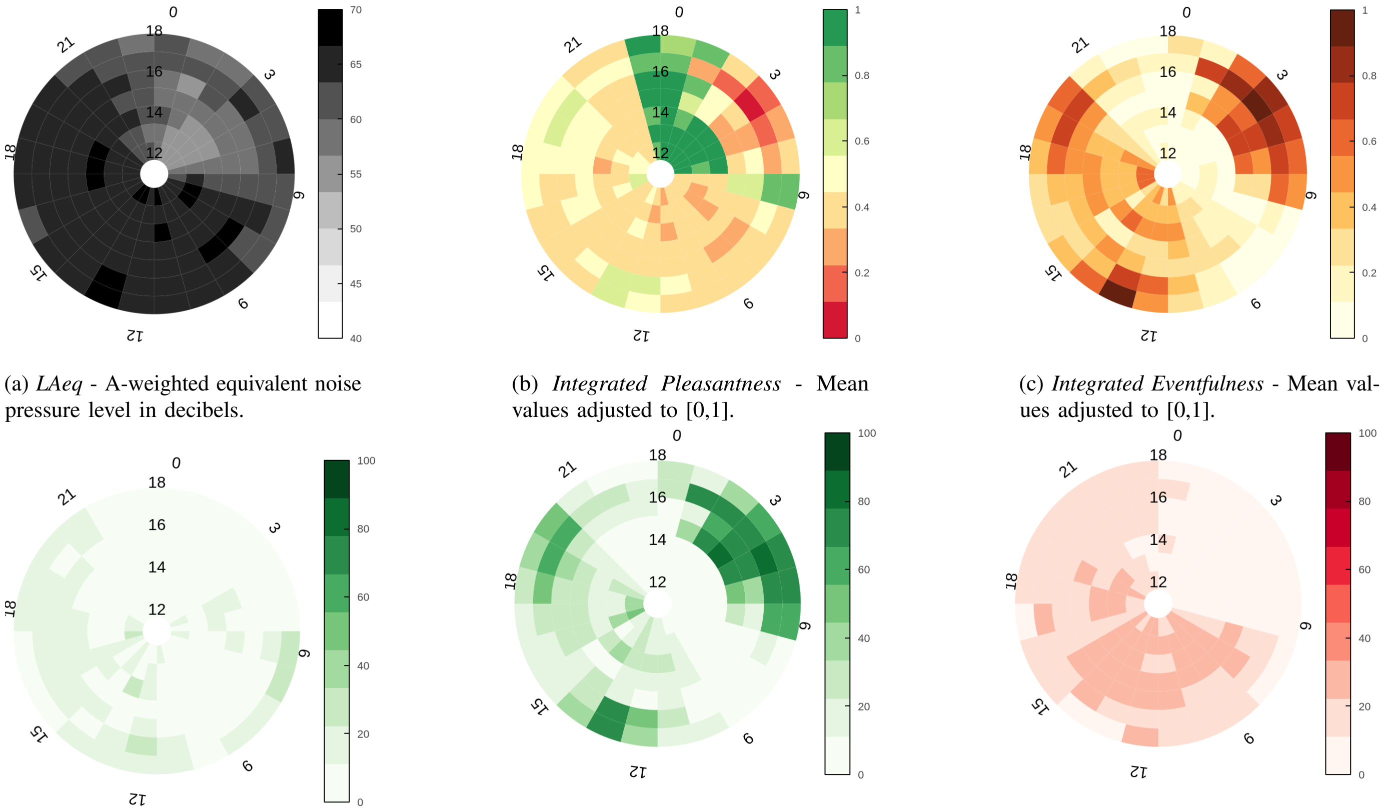


Fig. 2: SENS device deployed in the city of Pamplona/Iruña as part of the *Smart Iruña Lab* program.

In order to extract meaningful information from the raw monitored data, it is necessary to make it interpretable and practically useful by applying methodological choices (such as setting thresholds based on empirical testing to define when a sound source is considered active) and to aggregate data by statistical mean, the percentage of time above a threshold or the number of detected events. For example, Figure 3 presents an example of SENS monitoring results: the aggregated data by hour for the week of May 12-18 for one of the monitored sites in Pamplona/Iruña. In these plots, the 360 degrees represent the 24 hours of the day, while each concentric circle corresponds to a day of the week — with Monday at the center and Sunday on the outermost ring. The selected location is a residential neighborhood very close to the city center, known for its vibrant street life and frequent visits from young people due to nearby nightclubs. It is generally perceived as quite noisy because of constant traffic throughout the week.

Subplot (a) shows the *LAeq*. This confirms the high noise levels in the area, with L_{den} values ranging from 68 to 70 dB every day. To better understand the soundscape, it is useful to examine the perceptual attributes alongside the detected sound sources. However, first, it should be noted that due to the criteria of the city council, any activity at night (from 23:00 to 07:00) is inversely interpreted for *pleasantness*: the values for *pleasantness* during these hours are adjusted to reflect greater unpleasantness when activity is high, using the inverse of the *eventfulness* values. Looking at graph (b), we see that the day and evening periods tend to be neither distinctly pleasant nor unpleasant, likely because of the constant presence of traffic, as illustrated in graph (d). Nights in the second half of the week appear more unpleasant, which corresponds with higher activity levels (graph (c)) during the early morning hours from Thursday to Sunday. This pattern is explained by graph (e), which shows an increase in human presence at the same periods, indicating nightlife activity associated with the surrounding clubs and bars. Another notable aspect is the weekend activity peaks around 12:00–15:00 and 17:00–20:00. These high levels are also linked to human presence, suggesting that the area is a popular gathering spot during these times. The low detection



(a) *LAeq* - A-weighted equivalent noise pressure level in decibels.

(b) *Integrated Pleasantness* - Mean values adjusted to [0,1].

(c) *Integrated Eventfulness* - Mean values adjusted to [0,1].

(d) *Birds presence* - Time(%) over threshold, ($th = 0.15$ within the range $[0, 1]$).

(e) *Human presence* - Time(%) over threshold, ($th = 0.2$ within the range $[0, 1]$).

(f) *Vehicles presence* - Time(%) over threshold, ($th = 0.1$ within the range $[0, 1]$).

Fig. 3: Circular graphs of hourly data from May 12–18 for one site monitored in Pamplona/Iruña.

rates for birds (perceived 20–30% of the time) indicate that the area is highly urbanised, offering limited refuge for wildlife.

Altogether, this data illustrates how SENS enables us to understand the acoustic environment without needing to be physically present. By combining continuous measurement and intelligent data aggregation, we gain valuable insights into daily and weekly sound patterns. If noise issues arise, these insights provide robust evidence to identify the most effective measures to improve the urban soundscape.

7. CONCLUSION AND FUTURE WORK

The emergence of AI opens new opportunities for real-time sound analysis. The generally high cost of noise monitoring devices on-the-market raises the need to develop solutions based on small low-cost devices. Nevertheless, deploying AI-based sound analysis algorithms on real-time acoustic sensors presents numerous challenges, including the computational constraints linked with the need to protect data privacy. This paper has introduced SENS as a practical use case that addresses these difficulties through a low-cost, flexible, and privacy-preserving solution. By performing all audio processing locally on the device, SENS reduces the risk of compromising personal privacy. To achieve this, significant effort was made to reduce the computational load of the trained models through careful pruning and the use of Principal Component Analysis (PCA) for input dimensionality reduction. Further, the general software architecture is modular, with separate components for audio capture, signal processing, and transmission of analysis results to a remote server when required. Moreover, the methodology proposes the development of independent models for each acoustic parameter of interest, enhancing the system's adaptability and scalability. Internet data consumption is further optimized through batching and packaging multiple analysis results before transmission. To demonstrate its real-world viability, a network of SENS devices has been deployed in an urban environment. This long-term deployment has shown that the system is robust, reliable, and capable of generating meaningful insights into the acoustic environment.

Future work will focus on further optimizing and expanding the methodology by exploring the minimum viable sampling frequency needed for accurate predictions to help reduce processing load even further or using more lightweight protocols like MQTT for data transmission. Additionally, the modular design of SENS makes it well-suited for other applications beyond urban noise monitoring. Future applications could include traffic monitoring (e.g., counting vehicles or distinguishing between light and heavy traffic), public safety (e.g., detecting distress calls on the streets), or any other scenario where sound can serve as a valuable real-time input of information.

8. ACKNOWLEDGMENT

This work was supported by *IA y Música: Cátedra en Inteligencia Artificial y Música (TSI-100929-2023-1)* by *Secretaría de Estado de Digitalización e Inteligencia Artificial* and the European Union-Next Generation under *Cátedras ENIA 2022*, and the IMPA project *(PID2023-152250OB-I00)* by the Ministry of Science, Innovation and Universities of the Spanish Government, the *Agencia Estatal de Investigación (AEI)* and co-financed by the EU. We would also like to acknowledge *Keacoustic* for the hardware support and *Smart Iruña Lab* for the opportunity to deploy the system.

REFERENCES

[1] AudioMoth, https://www.openacousticdevices.info/audiomoth.

[2] D. Bonet-Solà, E. Vidaña-Vila, and R. M. Alsina-Pagès, “Acoustic comfort prediction: Integrating sound event detection and noise levels from a wireless acoustic sensor network,” *Sensors*, vol. 24, no. 13, p. 4400, Jul. 2024.

[3] CENSE project, https://cense.ifsttar.fr/en/.

[4] J. Ardouin, J.-C. Baron, L. Charpentier, D. Ecotière, N. Fortin, F. Gontier, G. Guillaume, M. Lagrange, G. Libouban, J. Picaut, and C. Ribeiro, “A high density network of low cost acoustic sensors based on wired and airborne transmission of spectral data,” *Euronoise*, 2021.

[5] M. Tailleur, V. Lostanlen, J.-P. Rivière, and P. Aumond, “Machine listening in a neonatal intensive care unit,” *DCASE*, 2024.

[6] P. Ngamthipwatthana, M. Götze, A. Kátai, and J. Abeßer, “Towards measuring and forecasting noise exposure at the veltins-arena in gelsenkirchen, germany,” *DCASE*, 2024.

[7] A. Sagasti, M. Rocamora, and F. Font, “Prediction of pleasantness and eventfulness perceptual sound qualities in urban soundscapes,” *DCASE Workshop*, 2024.

[8] ARAUS dataset, https://researchdata.ntu.edu.sg/dataset.xhtml?persistentId=doi:10.21979/N9/9OTEVX.

[9] ISO 12913-1. Acoustics-Soundscape-Part 1: Definition and conceptual framework, www.iso.org.

[10] ISO 12913-2. Acoustics-Soundscape-Part 2: Data collection and reporting requirements, www.iso.org.

[11] ISO 12913-3. Acoustics-Soundscape-Part 3: Data analysis, www.iso.org.

[12] Urban Sound Monitoring (USM) dataset, https://github.com/jakobabesser/USM.

[13] FSD50K Dataset, https://zenodo.org/records/4060432.

[14] IDMT-TRAFFIC Dataset, https://www.idmt.fraunhofer.de/en/publications/datasets/traffic.html.

[15] Urban Sound 8k dataset, https://urbansounddataset.weebly.com/urbansound8k.html.

[16] LAION-AI/CLAP Github Repository, https://github.com/LAION-AI/CLAP.

[17] R. O. Araz, D. Bogdanov, P. Alonso-Jiménez, and F. Font, “Evaluation of deep audio representations for semantic sound similarity,” *International Conference on Content-based Multimedia Indexing (CBMI)*, 2024.

[18] Smart Iruña Lab, https://www.pamplona.es/smart-iruna-lab.