Audio Engineering Society
Conference Paper 13
Presented at the AES International Conference on Machine Learning and Artificial Intelligence for Audio
2025 September 8-10, London, UK

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (http://www.aes.org/e-lib), all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

A Scalable AI Architecture for Audio and Multimodal Analysis on Mobile Devices: A Case of Environmental Monitoring

Marina Eirini Stamatiadou[1], Athanasia Mpesmerti[1], Nikolaos Vryzas[1], Lazaros Vrysis[1], Charalampos Dimoulas[1]

[1] *Multidisciplinary Media & Mediated Communication Research Group (M3C), Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*

ABSTRACT

The increasing need for real-time environmental monitoring has made Artificial Intelligence (AI) and Machine Learning (ML) models for audio classification and multimodal sensing essential for detecting and analyzing pollution-related sounds. In cases where mobile devices are used for capturing and processing audiovisual content, such models offer significant potential for research, storytelling, and public awareness. This study proposes a scalable and modular architecture that enables direct and flexible access to AI models for multimodal and audio processing. A CNN-LSTM hybrid model is trained for real-time environmental sound classification and deployed as a service. Building on prior 1D CNN-based approaches and incorporating temporal dependencies, the new model achieves an AUC of 0.91, demonstrating improved accuracy and generalization. The system leverages a REST API and Docker-based containerization, allowing deployment of independent AI services and supporting mobile and IoT use cases. The architecture accommodates both pre-trained and custom models, accessible from any device via a unified interface, confirming that a hybrid CNN-LSTM topology can support effective real-time sound classification, within a modular, containerized framework.

Keywords: Environmental Audio, CNN-LSTM, REST API, Multimodal, Microservices, Real-time AI

1. Introduction

1.1. Background and motivation

The rapid advancement of technology and the proliferation of Artificial Intelligence (AI) and Machine Learning (ML) models have opened the way to numerous applications that process audio, from speech recognition and music generation, to audio-event detection and environmental monitoring. In the context of environmental awareness, sound classification plays a crucial role in detecting sources of pollution, such as industrial noise, traffic congestion, and construction activities [1], [2]. Environmental sound monitoring (ESM) plays a crucial role in modern cities, where air and noise pollution affect public health and overall quality of life, as well as natural ecosystems, and is becoming a growing concern. Noise pollution exacerbates stress, anxiety, depression [3], hypertension, and a higher risk of cardiovascular diseases [4], [5], particularly in urban areas, where prolonged exposure disrupts sleep and learning, and is linked to cognitive impairments, especially for children near highways or construction zones [6]. Additionally, ESM contributes to indirect air pollution detection by analysing urban soundscapes and identifying acoustic footprints associated with major sources of air pollutants.

Although significant progress has been noted in audio processing algorithms, there are cases in which computational requirements, limited mobility or restricted public accessibility, prevent their

applicability and exploitation in real-world scenarios. The ubiquity of smartphones and other mobile devices presents an unprecedented opportunity for ESM. These devices offer significant advantages over traditional fixed monitoring stations, including mobility, computational capabilities, and widespread availability. Their integrated sensors (i.e., microphone, camera, GPS) enable comprehensive data collection, while on-device AI processing facilitates immediate noise classification. The contextual richness of mobile data collection integrates noise measurements with location data, temporal patterns, and environmental conditions, providing multidimensional and multi-modal insights. Additionally, mobile devices allow dual functionality, in both offline and online modes, a fact that makes them particularly valuable for continuous environmental monitoring across diverse settings—from densely populated urban centers to remote natural habitats.

Lately, several smart city implementations [7], [8], [9], exploit mobile-based noise monitoring solutions and address the need for extensive coverage through distributed citizen participation, rather than relying on limited fixed sensor networks. Their capability to provide (near) real-time analysis allow for immediate pollution detection and pollution levels exceedance, enabling prompt intervention measures. From an economic perspective, smartphones are a remarkably cost-effective solution compared to traditional monitoring systems that require stationary audio sensors, significantly reducing the infrastructure investment by leveraging existing personal devices. Another important aspect is that mobile platforms excel at dynamic tracking, offering the unique capability to monitor transient pollution sources such as temporary construction sites, public events, or changing traffic patterns that fixed sensors might miss entirely. The crowdsourcing dimension of mobile-based environmental monitoring transforms passive citizens into active environmental stewards. By engaging the public through accessible and user-friendly applications [10][11], this approach generates substantial datasets for comprehensive urban noise mapping while simultaneously raising environmental awareness. Sound patterns captured during environmental monitoring serve as valuable proxies for detecting air quality issues. The distinctive acoustic signatures are not only correlated with noise pollution but also indicate the presence of common air pollutants. This relationship enables environmental scientists to use audio data to identify potential air quality hotspots, effectively extending a single microphone's capabilities beyond noise measurement to provide indicators of broader environmental conditions.

Our research addresses the critical gap between advanced AI capabilities and practical field deployment, by developing a lightweight, accessible architecture that democratizes access to sophisticated environmental sound analysis. It builds upon previous work in environmental sound classification, where a one-dimensional Convolution Neural Network (1D CNN) classification model was employed to distinguish between audio footprints that **are** or **are not** related with air pollution [11], [12] (R/NR classes respectively). While the 1D CNN approach demonstrated promising classification performance, we extended our methodology by integrating a Long Short-Term Memory (LSTM) layer, enabling the system to capture temporal dependencies effectively and model sequential patterns in environmental audio data. The proposed classification model is transformed into an independently deployable microservice and leverages REST API technology to become accessible as a web service, enabling seamless integration and real-time environmental monitoring from any internet-connected mobile device. This approach eliminates traditional barriers to AI deployment by abstracting the technical complexity behind standardized API calls, allowing non-technical users to benefit from advanced audio processing capabilities without specialized knowledge. By creating this bridge between sophisticated ML models and everyday environmental monitors, we aim to significantly expand participation in pollution detection efforts.

1.2. Paper structure

The rest of the paper is structured as follows: in section 2 previous related work is presented, in order to give a comprehensive background in the underlying research fields. Section 3 outlines the methodology that has been followed for building and evaluating the overall system. Section 4 highlights the experimental process and the results that came out of it. Finally, section 5 discusses the conclusions of the research and opens the way for future work and further research in the field.

2. Related work

2.1 AI in Environmental Sound Monitoring

Urban noise classification has gained high attention in the last decades due to the increasing demand for accurate environmental monitoring within the context of pollution analysis and mitigation strategies.

Previous research has explored the use of Deep Learning models, especially CNNs and Recurrent Neural Networks (RNNs), for environmental sound classification [13]-[15]. Recent advancements in IoT-based urban noise monitoring exploit the integration of embedded systems like Raspberry Pi with deep learning models, leveraging data normalization, augmentation, and 2D CNN architectures to achieve high-accuracy real-time sound classification [13]. Additionally, Audio Entailment [17] enhances complex audio scene understanding, enabling multimodal fusion of audiovisual data with other environmental inputs for more accurate pollution reasoning.

In parallel, studies have demonstrated that hybrid deep learning models, such as CNN-LSTM architectures, achieve superior sound classification by combining feature extraction with temporal modelling [18]. Edge devices, like Jetson Nano, have been tested in wireless sensor networks, integrating MFCC and CNN-LSTM models to achieve low-latency, high-accuracy classification, addressing stability and real-time performance limitations in existing systems [19]. Meanwhile, an optimized Bayesian ensemble-based 1D-CNN has been tested for environmental sound classification, achieving state-of-the-art accuracy on raw waveforms with low computational cost [20]. Latest research leverages Sequential CNNs and VGG19 CNNs, which have demonstrated high accuracy in processing and analysing urban soundscapes, while adding an explainability layer using techniques such as Grad-CAM to enhance model interpretability, transparency, trust, and applicability in real-world noise monitoring applications [21], [22].

2.2 Microservices and REST APIs for AI Applications

A microservice-based architecture decomposes applications into independent services, each being responsible for a specific functionality and database. This software design approach allows modular development and easier maintenance [23], enhancing resilience by isolating faults within individual services. While microservices were initially deployed in virtual machines (VMs), they have evolved with containerization, which optimizes resource use and improves deployment efficiency [24]. Similar to common service-based architectures they also offer flexibility, enabling development in different programming languages and environments depending on specific needs and/or prior knowledge [25]. RESTful API-based communication is a scalable and robust approach for modern applications that facilitates easy integration with client applications, making machine learning and AI models more accessible to users, through a process known as *Restification*. This allows developers to train a model and expose it via HTTP requests (GET, POST, PUT, DELETE), enabling clients to send data and receive predictions.

In the context of environmental sound classification, this approach aligns with advancements in music information retrieval, where microservices have been successfully used to detect and analyze audio patterns, as seen in systems like MusicTalk [26]. A key advantage of deploying audio classification models as RESTful microservices is their ability to function independently while interacting with other services for multimodal environmental analysis. Studies in musical instrument recognition have shown how microservice-based architectures enable the integration of independent detectors while ensuring compatibility across datasets and applications. Similarly, in environmental sound monitoring, microservices allow multiple models—such as noise classification, air pollution inference from soundscapes, and event detection—to be dynamically linked, ensuring flexible and efficient system expansion [11], [12], [27], [28].

Building upon recent advances in both environmental sound classification and service-oriented architectures, our research synthesizes key insights from both domains. Real-time AI applications require efficient model architectures and optimized deployment strategies. Prior approaches have explored the use of edge computing and lightweight AI models, but these solutions often trade accuracy for speed. Our approach balances both by leveraging a CNN-LSTM hybrid architecture. As demonstrated in [29], encapsulating machine learning models as REST microservices removes significant implementation barriers while enhancing scalability. Similarly, the success of microservice architectures in music information retrieval systems like MusicTalk [26] provides a proven framework for audio pattern recognition which we have adapted for environmental audio classification and pollution monitoring. This convergence enables non-technical users to leverage sophisticated audio classification capabilities through standardized API calls, while maintaining the flexibility to incorporate new models as environmental science advances.

3. Methodology

The proposed system consists of a 1D-CNN-LSTM hybrid model to leverage both spatial and temporal

features in audio signals, in order to identify sources of air pollution. The developed model is encapsulated and containerized, deployed on the cloud, offered via an API as a REST microservice.

3.1 Dataset

The initial UrANP dataset, which contains audio clips related and not related to air pollution, was constructed using samples from the benchmark datasets UrbanSound8k and ESC-50 [27], [28]. These clips were manually selected and categorized into two distinct classes: *pollution-related (R)* and *non-pollution-related (NR)* sounds. The final dataset was formed by concatenating the selected clips into two separate waveform files, one for each class, using the Audacity software. This format allows the direct feeding of raw waveform input into the 1D CNN network that will be presented next. To further enhance the dataset, and analyze the model's performance across different dataset sizes, additional audio segments sourced from YouTube were incorporated into both categories, with durations of 1 hour, 30 minutes, 12 minutes, and 6 minutes, resulting in four training variations with total dataset durations of 10, 9, 8:24, and 8:12 hours.

3.2 Model training and optimization

The optimization of a classification model can range from modifications in the dataset to a complete restructuring of the training architecture. Built on the foundations of previous research [2], [11], [30] and further extended to optimize its accuracy, the 1D-CNN model is trained to classify environmental sounds in two distinct categories: related and not-related to air pollution sources (PR/NPR). Significant improvements in the output results of classifiers have been observed with the use of k-fold cross-validation [31], and the inclusion of data shuffling to enhance model generalization [32]. An alternative to k-fold cross-validation is described in the Scikit-learn library, which introduces the "ShuffleSplit" function. Unlike k-fold cross-validation, ShuffleSplit randomly selects training and validation subsets in each iteration, providing greater variability in data partitioning. Following a similar approach but within the context of k-fold cross-validation, we implement data shuffling before each partitioning step, ensuring a more balanced distribution of training samples across different folds, following the logic of stratified ShuffleSplit, as shown in Figure 1. Our optimization process includes the steps of hyperparameter configuration, k-fold cross validation and data shuffling.

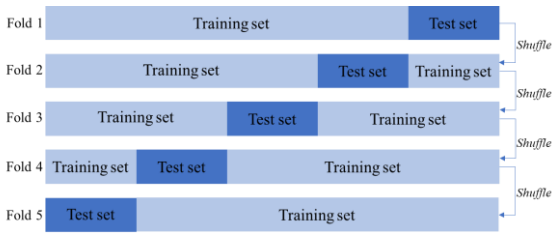


Figure 1: Data shuffling in 5-fold cross validation

3.2.1 1D-CNN classifier optimization

The optimization of the 1D-CNN classifier was attempted through deliberate manual hyperparameter tuning rather than automated methods, as the baseline model already showed promising accuracy. This decision was taken in order to ensure a balance between model performance and computational efficiency. Key parameters (i.e., CNN filters, kernel size, dropout rates) were selected and adjusted differently, analyzing each modification's effect on validation accuracy and potential overfitting. The hyperparameter optimization process confirmed that the initial configuration is the optimal one. The details are shown in Table 1.

Layer (type)	**Output shape**
Input Layer	(22050, 1)
Conv1D (F=32, CS=8)	(22043, 32)
MaxPooling1D (KS=8)	(2755, 32)
Dropout (10%)	(2755, 32)
Conv1D (F=64, KS=8)	(2748, 64)
MaxPooling1D (KS=8)	(343, 64)
Dropout (10%)	(343, 64)
Conv1D (F=96, KS=8)	(336, 96)
MaxPooling1D (KS=8)	(42, 96)
Dropout (10%)	(42, 96)
Conv1D (F=128, KS=8)	(35, 128)
MaxPooling1D KS=8)	(4, 128)
Dropout (10%)	(4, 128)
GlobalAveragePooling1D	(128,)
Dense (N=16, Fu:ReLU)	(16,)
Dropout (10%)	(16,)
Dense (N=2, Fu:Softmax)	(2,)

KS: Kernel Size, F: Filters, Fu: Function, N: Neurons
Table 1: 1D-CNN model architecture and hyperparameter configuration

Further enhancement of the model's generalization capabilities included k-fold cross-validation with additional fold configurations of k = 5 and k= 7, to

assess the optimal balance between accuracy's consistency and variance reduction. Best results were achieved for k=5 folds.

3.2.2 LSTM layer

To further optimize the model, a Long Short-Term Memory (LSTM) layer was added after the CNN, by removing the Global Average Pooling layer, as it reduces data dimensions and would prevent the LSTM from processing sequential dependencies. In a second training approach both Dropout and Max Pooling layers were removed, allowing the LSTM to be applied directly after the final Conv1D layer, preserving 35 timesteps. This approach retains more temporal information compared to maintaining Dropout and Max Pooling, which would have reduced the sequence length to 4. The final model architecture is presented in Table 2.

Layer (type)	**Output shape**
Input Layer	(22050, 1)
Conv1D (F=32, CS=8)	(22043, 32)
MaxPooling1D (KS=8)	(2755, 32)
Dropout (10%)	(2755, 32)
Conv1D (F=64, KS=8)	(2748, 64)
MaxPooling1D (KS=8)	(343, 64)
Dropout (10%)	(343, 64)
Conv1D (F=96, KS=8)	(336, 96)
MaxPooling1D (KS=8)	(42, 96)
Dropout (10%)	(42, 96)
Conv1D (F=128, KS=8)	(35, 128)
LSTM (64 units)	(64,)
Dropout (10%)	(64,)
Dense (N=16, Fu:ReLU)	(16,)
Dense (N=2, Fu:Softmax)	(2,)

KS: Kernel Size, F: Filters, Fu: Function, N: Neurons
Table 2: The hybrid CNN-LSTM model architecture

3.3 Restification

To enable real-time interaction with the trained classification model, a REST API was developed, using Flask, deploying the model as a service. The process includes Flask installation along with its dependencies, server configuration, routes and endpoints definition for the HTTP requests. Upon receiving an audio file, the system stores it in a temporary directory, allowing for immediate access during processing and ensuring file deletion after the prediction is completed to maintain system cleanliness.

1. For real-time handling, audio input is segmented into fixed-duration waveform chunks of 1 second with 0.5 seconds overlap, prior to inference. These buffered segments are forwarded to the classification model. This approach simulates real-time streaming and enables low-latency predictions even from non-continuous audio sources. The trained model is invoked via the "/predict" endpoint to classify the incoming audio as PR (pollution-related) or NPR (non-pollution-related), and the outcome is returned as a structured JSON response. In case of failure, the system includes robust error-handling routines to ensure reliability.

2. Finally, the service is configured to automatically launch and remain active, allowing users to send audio files for classification via a simple HTTP request. This architecture supports remote access, enabling real-time predictions from any connected device without requiring local model deployment, facilitating seamless integration into mobile and web-based applications.

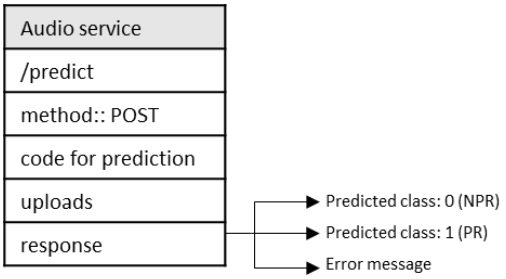


Figure 3: Audio classification model as a service

3.4 Integrated architecture with complementary services

Following the successful deployment of the hybrid audio classification model as a RESTful service, the next step expands the system to integrate additional models. For the needs of the current research, a topic modeling service (processing text) and an image event identification service (identifying pollution sources) were incorporated. To achieve seamless interoperability while preserving the modularity of each service, Docker containerization was employed, ensuring efficient management, scalability, and standardized execution across diverse computing environments. The first step in the containerization process is the selection of a base image, the definition of the workspace, copy and installation of the dependencies, and integration of the necessary model

files and application code. The ports are exposed to enable service and external communication.

In the final stage of the micro-service architecture implementation the three services were deployed on a shared cloud server, (Google Cloud), to achieve centralized management and secure access to each service through unique URLs that are automatically generated upon deployment. Cloud hosting ensures that these services remain accessible for real-time utilization of the models from the main application as well as any external interface.

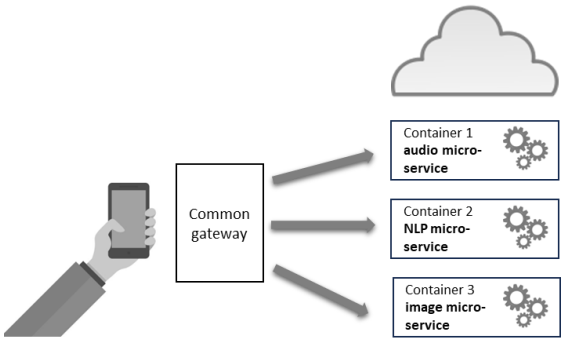


Figure 4: The proposed architecture

4. Experimental Results

The evaluation of the first model's training was conducted using precision, recall, F1-score, and accuracy, measured across each fold of k-fold cross-validation. These metrics quantify model performance, with precision assessing the accuracy of positive predictions, recall measuring the detection of actual positives, F1-score balancing both, and accuracy indicating overall correctness. The average accuracy across folds was reported, alongside the final fold's metrics for a representative performance snapshot. In cases of significant performance improvement, ROC-AUC analysis was performed using an independent dataset, visualizing model discrimination ability through True/False positive rate (TPR/FPR) calculations, with AUC measuring classification effectiveness.

Class	Precision	Recall	F1-score
0 (NPR)	0.77	0.75	0.76
1(PR)	0.76	0.77	0.77
Mean accuracy	**78.5 ±1.5%**		

Table 3: Baseline model evaluation

4.1 Model evaluation with manual hyperparameter configuration

Changes in hyperparameters and the overall architecture of the initial convolutional network resulted in significantly lower performance across evaluation metrics. Consequently, the original structure was retained, with subsequent optimization efforts focusing on specific aspects of the model to improve evaluation metrics.

4.2 Model evaluation with k-fold cross validation (k=5 and k=7)

In 5-fold cross-validation, performance slightly declined, with an average accuracy of 78.2% and 71.3% in the final fold, while variability increased (±4.5% deviation), indicating reduced model stability. In 7-fold cross-validation, accuracy further dropped to 76.4% overall and 70.1% in the final fold, with an even higher deviation (±6.1%), suggesting increased instability. These results indicate that increasing the number of folds did not enhance accuracy but rather introduced greater variability, confirming that 3-fold cross-validation was the most suitable choice for training the baseline model.

4.3 Model evaluation with shuffling

Using random shuffling (via numpy.random.shuffle function) for each fold significantly improved model performance, increasing accuracy from 78.5% to 92.5%. This enhancement indicates that random data redistribution strengthened the model's ability to generalize to new data, reducing overfitting and increasing sample diversity per fold. The large dataset size further supported generalization, allowing the model to learn more representative patterns and avoid dependence on specific data structures.

The results of data shuffling combined with 5-fold and 7-fold cross-validation showed that model accuracy remained consistently high at 93%, indicating only a slight improvement over 3-fold cross-validation. The stability of these results across different fold numbers suggests that the model successfully generalized due to the large dataset size and random data redistribution, which increased sample diversity in each fold.

However, although evaluation metrics showed a noticeable improvement, the AUC score of 0.87 (Figure 5) suggests that while the model fits the training data well, its ability to generalize remains uncertain. Using ROC-AUC analysis with an independent dataset, only a slight improvement was observed (AUC increased from 0.86 to 0.87),

indicating the higher reliability of ROC-AUC for performance evaluation, and the potential for further model optimization to enhance generalization.

	k=3		k=5		k=7	
Class	0	1	0	1	0	1
Precision	0.91	0.95	0.90	0.93	0.90	0.93
Recall	0.96	0.91	0.93	0.90	0.95	0.90
F1-Score	0.93	0.93	0.92	0.92	0.94	0.92
Mean Accuracy	92.5±0.5%		93±0.7%		93±0.7%	

Table 5: Model evaluation after data shuffling for k-fold cross validation, for k=3, 5, 7

4.4 Hybrid (CNN-LSTM) Model evaluation

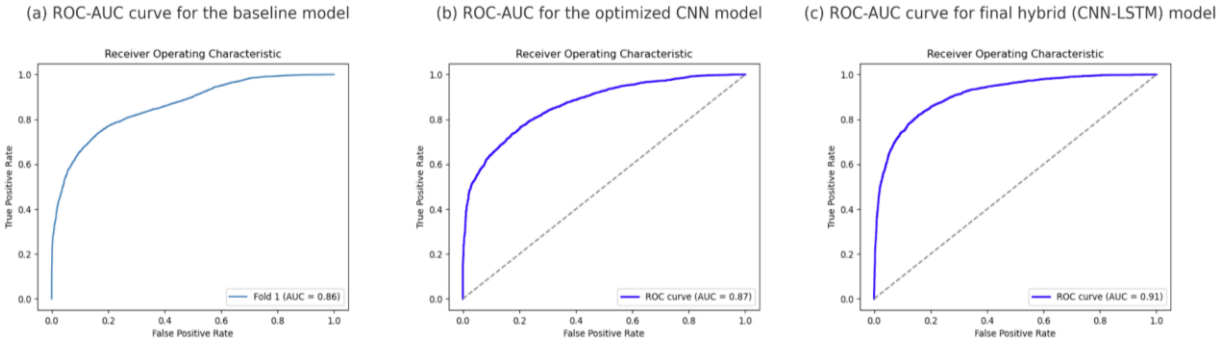
The final evaluation of the two LSTM-integrated training approaches showed consistently high model performance. In the first case (case a), where LSTM was placed immediately after the CNN, the model achieved 94.5% accuracy with 0.5% variability. In the second case (case b), where LSTM was added after the layer providing 35 timesteps, accuracy slightly improved to 95.3% with 0.3% variability. The results indicate that removing the Global Average Pooling layer allowed the LSTM to better utilize temporal information from the 35 timesteps, leading to enhanced performance. To ensure generalization and prevent overfitting, ROC curve was generated, by reserving 15% of the dataset, solely for validation.

The initial evaluation, conducted on a validation dataset derived from the training set, yielded an AUC score of 0.98, indicating potential overfitting. Although this dataset was not used during training, its small size and shared characteristics with the training data may have influenced the results, suggesting that the model memorized patterns rather than generalizing effectively.

To ensure a more objective assessment, a separate independent test-set was introduced. This dataset was sourced externally, processed with Audacity, and structured to include both pollution-related and non-pollution-related events, representing 20% of the total training data size. Using this dataset, the final ROC curve analysis produced an AUC score of 0.91, confirming that the model maintains high classification performance while demonstrating the ability to generalize to unseen data (Figure 6). The final ROC curve also indicates a high True Positive Rate (TPR) at low False Positive Rate (FPR) levels, verifying that the model achieves accurate predictions with minimal classification errors. These results validate the model's robustness and applicability in real-world environmental sound classification tasks.

	a		b	
Class	0	1	0	1
Precision	0.94	0.95	0.93	0.96
Recall	0.95	0.94	0.96	0.94
F1-Score	0.94	0.95	0.95	0.95
Mean Accuracy	**94.5 ± 0.5%**		**95.3 ± 0.3%**	

Table 6: Hybrid Model evaluation (a) LSTM layer right after CNN output (b) LSTM layer after last convolutional layer of the CNN (35 timesteps)



4.5 Micro-service architecture evaluation

The architecture's modular design allows for the incorporation of additional services, expanding its capabilities beyond audio classification. The integration of models for NLP and image event detection enables multi-modal data processing and analysis. This scalability enhances the system's applicability in environmental urban monitoring, allowing real-time analysis of diverse data sources while preserving the modularity and interoperability of the microservices framework.

To evaluate the proposed microservice-based architecture a prototype application was designed and developed, enabling real-time interaction with the deployed models. A web-based interface was

provided, allowing users to upload audio files and receive classification results via the integrated RESTful microservices. The assessment focused on system usability and accessibility, ensuring that the architecture effectively supported real-time environmental sound classification while maintaining flexibility for future extensions.

To measure the usability of the system, a small-scale System Usability Scale (SUS) test was performed, with a sample of 10 people with diverse technical backgrounds (journalists, engineers, communication experts, audio experts) [33]. The obtained average SUS score of 81,88 is well above the average usability threshold of 68 and indicates a positive user experience, confirming that the architecture

provides efficient, accessible, and intuitive interaction with the classification models. It is noteworthy to say that one of the key positive aspects noted was the ability to access the service directly via mobile devices, reinforcing its practicality in real-world scenarios. These results validate the potential of a microservices-based approach in AI-driven environmental monitoring, demonstrating its ability to provide scalable and modular processing capabilities while maintaining a user-friendly interface.

Additionally, the proposed approach has been tested in real-world conditions during three pilot deployments of the SOCIO-BEE project, which aims to engage citizens in environmental monitoring. In these pilots, participants used mobile devices to capture urban soundscapes in areas of interest. They followed the suggested process of uploading audio samples, receiving classification results, and contextualizing the audio with semantic descriptions. These semantic annotations were combined with real-time air quality sensor measurements, enabling a richer interpretation of environmental conditions. This use case highlights the potential of the proposed architecture to support citizen science, pollution awareness, and data enrichment in sustainable urban planning initiatives.

5. Discussion

The selected architecture for deployment is a CNN-LSTM model, where the LSTM layer is placed immediately after the final convolutional layer. This configuration allows the LSTM to fully leverage temporal dependencies extracted by the CNN while preserving the maximum timesteps available from the last convolutional layer. Additionally, data shuffling was applied per fold, increasing training sample diversity and further improving model generalization. The final evaluation was conducted using a 5-fold cross-validation strategy. A comparison between the ROC-AUC curves of the baseline model and the final optimized model clearly demonstrates the superiority of the latter. The increase in AUC from 0.86 to 0.91 validates the improvements in both architecture and training methodology, confirming enhanced classification performance and reliability on independent data. This makes the final model more suitable for real-world applications.

Beyond model performance, the study highlights the importance of architectural design in AI-driven environmental monitoring systems. The integration of RESTful microservices and Docker-based containerization allowed each model—audio classification, NLP, and image event identification—to operate independently while remaining accessible through a unified system. This modular approach ensures scalability and adaptability, making it possible to integrate additional models for multi-modal data processing, without significant architectural modifications.

While the core CNN-LSTM model architecture constitutes an extension of prior work, the primary contribution of this study lies in the development of an end-to-end, audio-focused AI infrastructure. The system enables real-time environmental sound classification through a modular, REST-based architecture that supports deployment as lightweight services, accessible via mobile or web interfaces. This allows for direct inference from raw audio waveforms, without requiring pre-segmented inputs, and facilitates integration with multi-modal data sources such as images or text metadata. The combination of raw waveform ingestion, cloud-based scalability, and mobile compatibility makes the proposed architecture particularly suitable for audio-centric applications in environmental monitoring. Future advancements could focus on container orchestration (e.g., use of Kubernetes for automated service scaling) and task distribution mechanisms (e.g., Celery task queue for asynchronous processing), to optimize resource allocation and improve responsiveness in real-time applications, addressing the growing demands of AI-driven audiovisual environmental monitoring.

References

[1] M. E. Stamatiadou, V. Nikolaos, V. Lazaros, and C. Dimoulas, "Identifying Sources of Air Pollution from Their Audio Footprints," in *Hellenic Institute of Acoustics*, Thessaloniki, Oct. 2022.

[2] A. Emvoliadis, N. Vryzas, M.-E. Stamatiadou, L. Vrysis, and C. Dimoulas, "Multimodal Environmental Sensing Using AI & IoT Solutions: A Cognitive Sound Analysis Perspective," *Sensors*, vol. 24, no. 9, p. 2755, 2024.

[3] J. Hegewald *et al.*, "Traffic noise and mental health: a systematic review and meta-analysis," *Int. J. Environ. Res. Public. Health*, vol. 17, no. 17, p. 6175, 2020.

[4] T. Münzel, M. Sørensen, and A. Daiber, "Transportation noise pollution and cardiovascular disease," *Nat. Rev. Cardiol.*, vol. 18, no. 9, pp. 619–636, 2021.

[5] T. Münzel, T. Gori, W. Babisch, and M. Basner, "Cardiovascular effects of environmental noise exposure," *Eur. Heart J.*, vol. 35, no. 13, pp. 829–836, 2014.

[6] J. Banks and D. Fink, "Noise as a Public Health Hazard," *The Hearing Journal*, vol. 75, no. 5. LWW, p. 6, 2022.

[7] A. Longo, M. Zappatore, M. Bochicchio, and S. B. Navathe, "Crowd-sourced data collection for urban monitoring via mobile sensors," *ACM Trans. Internet Technol. TOIT*, vol. 18, no. 1, pp. 1–21, 2017.

[8] B. Peng, K. I.-K. Wang, and W. H. Abdulla, "An Integrated Hierarchical Wireless Acoustic Sensor Network and Optimized Deep Learning Model for Scalable Urban Sound and Environmental Monitoring," *Appl. Sci.*, vol. 15, no. 4, p. 2196, 2025.

[9] B. İşler, "Urban Sound Recognition in Smart Cities Using an IoT–Fog Computing Framework and Deep Learning Models: A Performance Comparison," *Appl. Sci.*, vol. 15, no. 3, p. 1201, 2025.

[10] M. E. Stamatiadou, I. Thoidis, N. Vryzas, L. Vrysis, and C. Dimoulas, "Semantic crowdsourcing of soundscapes heritage: a mojo model for data-driven storytelling," *Sustainability*, vol. 13, no. 5, p. 2714, 2021.

[11] M. E. Stamatiadou, N. Vryzas, L. Vrysis, T. Saridou, and C. Dimoulas, "A citizen science approach to support joint air quality and noise monitoring in urban areas," in *Audio Engineering Society Convention 152*, 2022.

[12] M. E. Stamatiadou, N. Vryzas, L. Vrysis, and C. Dimoulas, "Beemate the game: A hunting treasure serious game for raising awareness and audience engagement in air pollution monitoring," in *2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)*, IEEE, 2023, pp. 1–4.

[13] Demir, F., Abdullah, D. A., & Sengur, A. (2020). A new deep CNN model for environmental sound classification. *IEEE Access*, *8*, 66529-66537.

[14] K. Zaman, M. Sah, C. Direkoglu, and M. Unoki, "A survey of audio classification using deep learning," *IEEE Access*, vol. 11, pp. 106620–106649, 2023.

[15] Z. Mushtaq, S. F. Su, and Q. V. Tran, "Spectral images based environmental sound classification using CNN with meaningful data augmentation," *Appl. Acoust.*, vol. 172, p. 107581, 2021.

[16] S. K. Shah, Z. Tariq, and Y. Lee, "Iot based urban noise monitoring in deep learning using historical reports," in *2019 IEEE International Conference on Big Data (Big Data)*, IEEE, 2019, pp. 4179–4184.

[17] S. Deshmukh *et al.*, "Audio Entailment: Assessing Deductive Reasoning for Audio Understanding." 2024. [Online]. Available: https://arxiv.org/abs/2407.18062

[18] O. O. Abayomi-Alli, R. Damaševičius, A. Qazi, M. Adedoyin-Olowe, and S. Misra, "Data Augmentation and Deep Learning Methods in Sound Classification: A Systematic Review," *Electronics*, vol. 11, no. 22, 2022, doi: 10.3390/electronics11223795.

[19] B. Peng, W. H. Abdulla, I. Kevin, and K. Wang, "Urban noise monitoring using edge computing with CNN-LSTM on Jetson Nano," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2023, pp. 2244–2250.

[20] M. G. Ragab, S. J. Abdulkadir, N. Aziz, H. Alhussian, A. Bala, and A. Alqushaibi, "An ensemble one dimensional convolutional neural network with Bayesian optimization for

environmental sound classification," *Appl. Sci.*, vol. 11, no. 10, p. 4660, 2021.

[21] M. Agarwal, K. S. Gill, S. Chattopadhyay, and M. Singh, "Classification of Urban Sound using Sequential Convolutional Neural Network (CNN) Model and its Visualisation," in *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, IEEE, 2024, pp. 1–5.

[22] M. Agarwal, K. S. Gill, P. Aggarwal, R. S. Rawat, and G. Sunil, "Urban Sound Classification using VGG19 Convolutional Neural Network (CNN) Model and its Visualisation," in *2024 4th International Conference on Innovative Practices in Technology and Management (ICIPTM)*, IEEE, 2024, pp. 1–5.

[23] J. Lewis and M. Fowler, "Microservices: a definition of this new architectural term," *MartinFowler Com*, vol. 25, no. 14–26, p. 12, 2014.

[24] D. Merkel and others, "Docker: lightweight linux containers for consistent development and deployment," *Linux J*, vol. 239, no. 2, p. 2, 2014.

[25] E. Wolff, *Microservices: flexible software architecture*. Addison-Wesley Professional, 2016.

[26] Y.-B. Lin, C.-C. Cheng, and S.-C. Chiu, "MusicTalk: A Microservice Approach for Musical Instrument Recognition," *IEEE Open J. Comput. Soc.*, vol. 5, no. 01, pp. 612–623, Jan. 2024, doi: 10.1109/OJCS.2024.3476416.

[27] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. 23rd ACM Int. Conf. Multimedia*, Brisbane, Australia, Oct. 2015, pp. 1015–1018.

[28] J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proc. 22nd ACM Int. Conf. Multimedia*, Orlando, FL, USA, Nov. 2014, pp. 1041–1044.

[29] M.-O. Pahl and M. Loipfinger, "Machine learning as a reusable microservice," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 2018, pp. 1–7. doi: 10.1109/NOMS.2018.8406165.

[30] L. Vrysis, I. Thoidis, C. Dimoulas, and G. Papanikolaou, "Experimenting with 1D CNN architectures for generic audio classification," in *Proc. Audio Eng. Soc. (AES) Conv. 148*, May 2020.

[31] S. Widodo, H. Brawijaya, and S. Samudi, "Stratified K-fold cross validation optimization on machine learning for prediction," *Sink. J. Dan Penelit. Tek. Inform.*, vol. 6, no. 4, pp. 2407–2414, 2022.

[32] J. Brownlee, "A gentle introduction to k-fold cross-validation," *Mach. Learn. Mastery*, vol. 2019, 2018.

[33] J. Brooke and others, "SUS-A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.