future internet

MDPI

Article

Empowering Healthcare: TinyML for Precise Lung Disease Classification

Youssef Abadade [1], Nabil Benamar [2,3], Miloud Bagaa [4,*] and Habiba Chaoui [1]

[1] System Engineering Laboratory, National School of Applied Siences, Ibn Tofail University of Kenitra, Kenitra B.P 242, Morocco; youssef.abadade@uit.ac.ma (Y.A.); habiba.chaoui@uit.ac.ma (H.C.)
[2] School of Technology, Moulay Ismail University of Meknes, Meknes 50050, Morocco; n.benamar@umi.ac.ma
[3] School of Science and Engineering, Al Akhawayn University in Ifrane, P.O. Box 104, Hassan II Avenue, Ifrane 53000, Morocco
[4] Department of Electrical and Computer Engineering, University of Quebec at Trois-Rivieres, Trois-Rivieres, QC G8Z 4M3, Canada
* Correspondence: miloud.bagaa@uqtr.ca

Abstract: Respiratory diseases such as asthma pose significant global health challenges, necessitating efficient and accessible diagnostic methods. The traditional stethoscope is widely used as a non-invasive and patient-friendly tool for diagnosing respiratory conditions through lung auscultation. However, it has limitations, such as a lack of recording functionality, dependence on the expertise and judgment of physicians, and the absence of noise-filtering capabilities. To overcome these limitations, digital stethoscopes have been developed to digitize and record lung sounds. Recently, there has been growing interest in the automated analysis of lung sounds using Deep Learning (DL). Nevertheless, the execution of large DL models in the cloud often leads to latency, dependency on internet connectivity, and potential privacy issues due to the transmission of sensitive health data. To address these challenges, we developed Tiny Machine Learning (TinyML) models for the real-time detection of respiratory conditions by using lung sound recordings, deployable on low-power, cost-effective devices like digital stethoscopes. We trained three machine learning models—a custom CNN, an Edge Impulse CNN, and a custom LSTM—on a publicly available lung sound dataset. Our data preprocessing included bandpass filtering and feature extraction through Mel-Frequency Cepstral Coefficients (MFCCs). We applied quantization techniques to ensure model efficiency. The custom CNN model achieved the highest performance, with 96% accuracy and 97% precision, recall, and F1-scores, while maintaining moderate resource usage. These findings highlight the potential of TinyML to provide accessible, reliable, and real-time diagnostic tools, particularly in remote and underserved areas, demonstrating the transformative impact of integrating advanced AI algorithms into portable medical devices. This advancement facilitates the prospect of automated respiratory health screening using lung sounds.

Keywords: TinyML; lung disease classification; early detection

Citation: Abadade, Y.; Benamar, N.; Bagaa, M.; Chaoui, H. Empowering Healthcare: TinyML for Precise Lung Disease Classification. *Future Internet* **2024**, *16*, 391. https://doi.org/10.3390/fi16110391

Academic Editors: Yuezhi Zhou and Xu Chen

Received: 24 August 2024
Revised: 17 October 2024
Accepted: 21 October 2024
Published: 25 October 2024

1. Introduction

According to the World Health Organization (WHO), lung diseases are among the leading causes of mortality worldwide, resulting in the deaths of millions of people each year [1]. Respiratory diseases are often detected late, making treatment less effective [2].

Various clinical approaches have been developed to diagnose and assess lung health issues, including computed tomographic scans, chest X-rays, and pulmonary function tests. However, these techniques are restricted to specialized medical facilities due to their complexity, high cost, and time-consuming nature [3]. Additionally, medical professionals in hospitals are often overworked, which increases the likelihood of errors and patient waiting times [3]. Therefore, it becomes apparent that a different approach is needed to better assist practitioners in making an initial diagnosis.

In contrast, the stethoscope is used as a non-invasive and patient-friendly tool for diagnosing respiratory conditions through lung auscultation [4]. This procedure involves listening to the sounds produced by air moving in and out of the lungs.

During lung auscultation, experts are able to identify various abnormal respiratory sounds, like wheezing and crackling [4]. These sounds serve as indicators of possible respiratory conditions for the patient. However, traditional stethoscopes come with several associated challenges. Firstly, their effectiveness relies heavily on the physician's expertise and judgment, introducing potential for diagnostic errors [5]. Secondly, they lack a recording feature, preventing other medical personnel from analyzing the sounds heard during consultations [6]. Thirdly, they are not equipped with noise-canceling capabilities, making it difficult to hear lung sounds in noisy environments such as emergency rooms or busy clinics [7].

The digital stethoscope has introduced a new approach to auscultation, benefiting research, education, and clinical practice [8]. It digitizes lung sounds, allowing for recording and playback, which reduces reliance on a single physician's judgment and enables collaboration with other medical professionals [8]. It incorporates digital filters to eliminate noise and isolate the relevant acoustic signals within specific frequency bands [8]. This enhances diagnostic accuracy and improves clinical decision making. It also allows for the visualization and retrospective analysis of lung sounds. The integration of wireless transmission capabilities, such as Bluetooth or WiFi, with the digital stethoscope will facilitate remote diagnosis, greatly enhancing convenience and application in a variety of medical contexts [8].

In recent years, there has been growing interest in the automated analysis of lung sounds. By using machine learning, particularly Deep Learning (DL) techniques, the experience, quality of diagnosis, and care for both patients and healthcare professionals have significantly improved [3,8]. The utilization of DL algorithms to examine lung sound patterns captured by digital stethoscopes represents a promising approach for the early and precise detection of disease [9]. Moreover, these technologies aim not only to reduce dependency on specialist facilities but also to overcome the limitations of traditional stethoscopes, making diagnosis more accurate by removing human error [3].

However, coupling digital stethoscopes with DL presents certain limitations. DL algorithms require significant computational resources, posing challenges in resource-constrained environments [10,11]. Latency in cloud-based solutions can impact real-time analysis, particularly in areas with insufficient internet bandwidth for large data transmission [10–12], and privacy concerns arise when transmitting sensitive health data over the internet [10,11]. To address these limitations, Tiny Machine Learning (TinyML) [13] offers a compelling solution by enabling efficient ML codes to run on small, energy-efficient devices.

TinyML is a fast-growing field of ML including hardware, algorithms, and software that aims to facilitate running ML models on ultra-low-power devices having very limited power (under 1 mW), less memory, and limited processor capabilities [14]. TinyML offers tiny IoT devices the ability to analyze data collected by various sensors and act based on the decisions made by the embedded ML model without the need for the cloud. TinyML finds applications in diverse fields [11], including agriculture [11,15], healthcare [10,11,16], and environmental monitoring [11,17].

The hardware limitations of tiny IoT devices require the minimizing of the ML model in order to deploy it in extremely resource-limited devices. The minimization of the model can be performed by the following techniques: pruning and quantization [14]. These techniques aim to reduce the size of the ML model while trying not to impact its accuracy. The pruning technique is the process of removing unused weights in the model to increase speed inference and minimize its size, while quantization reduces the precision of the model parameters from floating-point (e.g., 32-bit) to lower (e.g., 8-bit) precision [11]; this decreases the model's memory footprint as well as the amount of processing required.

TinyML holds immense potential in the healthcare sector [10,11,16]. TinyML's ability to run directly on devices at the edge offers numerous advantages. One of the most

significant benefits is the ability to perform real-time data analysis without the need for continuous data transmission to centralized cloud systems [11,16,18]. This reduces latency, enhances data privacy by minimizing the transfer of sensitive patient data, and lowers dependency on reliable internet connections [10,16], which is particularly beneficial for remote health monitoring in remote and undeserved areas [12,16].

Many studies have demonstrated the practical applications of TinyML in healthcare, showcasing its potential to optimize real-time health monitoring and diagnostic tools. The authors of [19] optimized a Convolutional Neural Network (CNN) model through pruning and quantization, making it deployable on low-cost microcontrollers like the Raspberry Pi Pico and ESP32 for real-time blood pressure estimation using photoplethysmogram (PPG) signals. This approach enables efficient, low-power solutions for continuous blood pressure monitoring. The authors of [20] proposed a TinyML-based solution for predicting and detecting falls among elderly individuals, utilizing a wearable device placed on the leg to capture movement data. The system employs a nonlinear support vector machine classifier for real-time fall detection and prediction. Similarly, the authors of [21] used CNN models combined with audio data to detect falls. In [22], researchers developed a system that predicts blood glucose levels in individuals with type 1 diabetes by deploying DL models on edge devices. This system, which relies on recurrent neural networks (RNNs), processes continuous glucose monitoring (CGM) data on low-power, resource-constrained devices, enabling real-time monitoring without the need for cloud infrastructure. Additionally, the authors of [23] introduced a lightweight solution based on Temporal Convolutional Networks (TCNs) for heart rate estimation in wearable devices. By leveraging optimized TCN models, the system achieved accurate heart rate monitoring while maintaining low latency and energy consumption, making it suitable for use in resource-constrained environments like wearable health devices.

In this paper, we present a new approach focused on creating TinyML models to distinguish between asthma and non-asthma conditions by using lung sound recordings. We developed and compared various ML models based on different metrics. To ensure these models are suitable for cost-effective platforms such as the Arduino Nano 33 BLE, we employed quantization techniques.

The remainder of the paper is organized as follows: Section 2 reviews existing studies that have addressed similar challenges. Section 3 details the materials and methodologies used in our experiments. Section 4 provides a comprehensive analysis of the empirical results and their implications. Finally, Section 5 presents our conclusions and proposes directions for future research.

2. Related Works

Numerous research papers have examined the application of DL to identify patterns and distinguish among various lung conditions by using raw respiratory sound data.

The authors of [24] developed a framework for classifying various respiratory diseases by using lung sound recordings. They employed a CNN with Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction. The proposed model achieved a classification accuracy of 95.7%, effectively distinguishing among different respiratory diseases, such as asthma, COPD, URTI, LRTI, and bronchiectasis, and a class representing healthy people.

In [25], the authors developed a framework for classifying lung sounds, addressing the challenge of noise interference from the heart and lung sounds. By utilizing two public datasets with 280 lung sounds of varying durations and sampling rates, the study preprocessed signals for uniformity and employed Mel-Frequency Cepstral Coefficients (MFCCs) [26] and Short-Time Fourier Transform (STFT) [27] for feature extraction. It tested several models, achieving the highest accuracy with an STFT+MFCC-ANN combination, demonstrating promising results for automatic respiratory diagnosis with high precision and recall rates.

In the paper [5], the authors evaluated the efficacy of various deep learning models for diagnosing respiratory pathologies by using lung auscultation sounds. The study

compared three deep learning models across both non-augmented and augmented datasets, revealing that the CNN–LSTM model outperformed others with high accuracy rates in all scenarios. Augmentation significantly enhances model performance, with the CNN–LSTM hybrid showing particular strength by combining the CNN's feature extraction capabilities with LSTM's [28] classification efficiency.

In [29], the authors explored the effectiveness of Mel Frequency Cepstral Coefficients (MFCCs) in classifying cough sounds for diagnosing five respiratory diseases, such as asthma and chronic obstructive pulmonary diseases (COPDs). The study employed a unique ensemble of recurrent neural network models with LSTM cells and tested various meta-classifiers, achieving over 87% accuracy. This approach underscores MFCCs' potential as standalone features for cough signal classification and suggests future directions, including further disease characteristic diagnosis and COVID-19 cough classification.

In [30], the authors tackled the challenge of detecting respiratory pathologies from sounds, using the ICBHI Benchmark dataset. Given the dataset's imbalance, the study employed a Variational Convolutional Autoencoder for data augmentation, alongside traditional oversampling techniques. A CNN was employed for classification into healthy, chronic, and non-chronic categories, achieving an F-score of 0.993 in the three-label classification. For the six-class classification, which included RTI, COPD, Bronchiectasis, Pneumonia, and Bronchiolitis, the CNN achieved an F-score of 0.99.

In [31], the authors developed a non-invasive method for classifying respiratory sounds by using an electronic stethoscope and audio recording software. By using MFCC with SVM and spectrogram images with CNN, they benchmarked the CNN's performance against the SVM method across various sound classifications. The CNN and SVM both reached 86% in distinguishing healthy versus pathological sounds; for rale, rhonchus, and normal sounds, the CNN achieved 76% and SVM 75%; in singular-sound-type classification, both achieved 80%. These results underline the effectiveness of CNNs and SVM in respiratory sound analysis.

In [32], the authors developed a system for diagnosing asthma using deep learning by analyzing respiratory sounds from asthmatic and non-asthmatic individuals. They developed a web interface and a mobile app for real-time prediction, aiding doctors in performing accurate diagnoses. Utilizing features such as chroma, RMS, Spectral centroid, Rolloff, and MFCCs, the ConvNet model demonstrated impressive performance metrics, including 99.8% accuracy, 100% sensitivity, 100% specificity, and a 99% F-score.

In [33], the authors proposed RDsLINet, a novel lightweight inception network for classifying a broad spectrum of respiratory diseases through lung sound signals. The framework involves preprocessing, melspectrogram image conversion, and classification via RDsLINet. The proposed RDsLINet achieved impressive accuracy rates: 96% for seven-class, 99.5% for six-class, and 94% for healthy vs. asthma classifications.

In [34], the authors proposed a novel approach to respiratory disease detection through a wearable auscultation device. They developed a Respiratory Sound Diagnosis Processor Unit (RSDPU) utilizing LSTM networks to analyze respiratory sounds in real time. The study highlights the implementation of Dynamic Normalization Mapping (DNM) to optimize quantization and reduce overfitting, crucial to maintaining model accuracy with limited computational resources. The hardware implementation of the RSDPU includes a noise corrector to enhance diagnostic reliability. The results show that the RSDPU achieved an 81.4% accuracy in abnormality diagnosis, with a minimal power consumption of 381.8 μW. The study demonstrates the potential of combining advanced machine learning techniques with efficient hardware design to create effective and practical healthcare solutions for continuous respiratory monitoring

A startup called Respira Labs [35] has introduced an innovative, cost-effective wearable sensor that leverages TinyML to analyze cough sounds for signs of respiratory diseases like pneumonia. This compact device integrates a microphone and a microcontroller executing a neural network to discern specific cough characteristics such as wheezing and

crackling. Designed for ease of use, it features a simple strap mechanism, operates without batteries, and communicates results via LED indicators and sound signals.

Table 1 summarizes existing research on lung disease detection and classification using audio data, covering diseases such as asthma, COPD, lung fibrosis, bronchitis, and pneumonia and various pathological lung sounds. Datasets vary from publicly available ICBHI 2017 to self-collected data, indicating diversity in data sources. The extracted audio features include STFT, MFCC, and spectrograms, with MFCC being the most common. The models used range from ANN, CNN, and LSTM, to hybrid CNN-LSTM, achieving high accuracy rates and mostly deploying solutions on cloud platforms. While some studies [33,34] have explored computation directly on edge devices, our work leverages TinyML to push the boundaries of what can be achieved on resource-limited hardware. This approach allows for efficient real-time analysis and model deployment on compact, low-power devices, making advanced diagnostics more accessible in a wide range of settings.

Table 1. Related work summary.

Work	Lung Diseases	Dataset	Feature(s)	Model	Results	Deployment
[24]	Asthma, COPD, URTI, LRTI, and bronchiectasis and normal	Respiratory Sound Database	MFCC	CNN	95.7%	Cloud
[25]	Normal and abnormal	ICBHI 2017 and KAUH	STFT and MFCC	ANN, SVM, KNN, DT, and RF	ANN: 98.61%; SVM: 93%; KNN: 93%; DT: 88%; RF: 95%	Cloud
[5]	Asthma, BRON, COPD, heart failure, lung fibrosis, normal, and pleural effusion pneumonia	ICBHI 2017 and KAUH	-	CNN and LSTM and CNN-LSTM	CNN: 99.81%; LSTM: 99.81%; CNN-LSTM: 100%	Cloud
[29]	Asthma, COPD, ILD, bronchitis, and pneumonia	Self-collected data	MFCC	LSTM	88.5%	Cloud
[30]	Healthy, chronic, and non-chronic	ICBHI 2017	MFCC	CNN	Accuracy: 99%	Cloud
[31]	Healthy and pathological	Self-collected data	MFCC and spectrogram	SVM-MFCC, CNN-Spectrogram	MFCC-SVM: 86%; Spectrogram-CNN: 86%	Cloud
[31]	Rale, rhonchus and normal	Self-collected data	MFCC and spectrogram	SVM-MFCC, CNN-Spectrogram	MFCC-SVM: 76%; Spectrogram-CNN: 76%	Cloud
[32]	Asthma and non-asthma	Self-collected data	Chroma, RMS, and MFCC	ConvNet	Accuracy: 99.8%	Mobile
[33]	Healthy and asthma	KAUH	Mel-spectrogram	RDsLINet	Accuracy: 94%	Edge
[33]	Asthma, BRON, COPD, pneumonia, heart failure, and pleural effusion	KAUH	Mel-spectrogram	RDsLINet	Accuracy: 94%	Edge
[34]	Normal and unnormal	ICBHI 2017	MFCC	LSTM	Accuracy: 81.4%	Edge

3. Materials and Methods

In this section, we describe the various methods employed to acquire, preprocess, and build models for distinguishing between normal and asthma conditions. The development process was conducted by using Edge Impulse, a platform specifically designed for training models, adjusting hyperparameters, and optimizing them for operation on various edge devices. Figure 1 depicts our approach, which involves the standard stages found in traditional machine learning projects, with added steps for real-time processing on the microcontroller:

Step 1: Data collection and preparation: Lung sound data were acquired from a publicly accessible dataset, divided into 80% for training and 20% for testing. These data were then processed by using bandpass filtering to reduce external noise.

Step 2: The lung sound data were uploaded to the Edge Impulse platform [36] and segmented into 5-second windows, with MFCC features extracted from each window.

Step 3: A custom CNN, a CNN proposed by Edge Impulse, and a custom LSTM were created, trained, and optimized for deployment on a microcontroller.

Step 4: The three models were tested by using the test data, and performance metrics were computed.

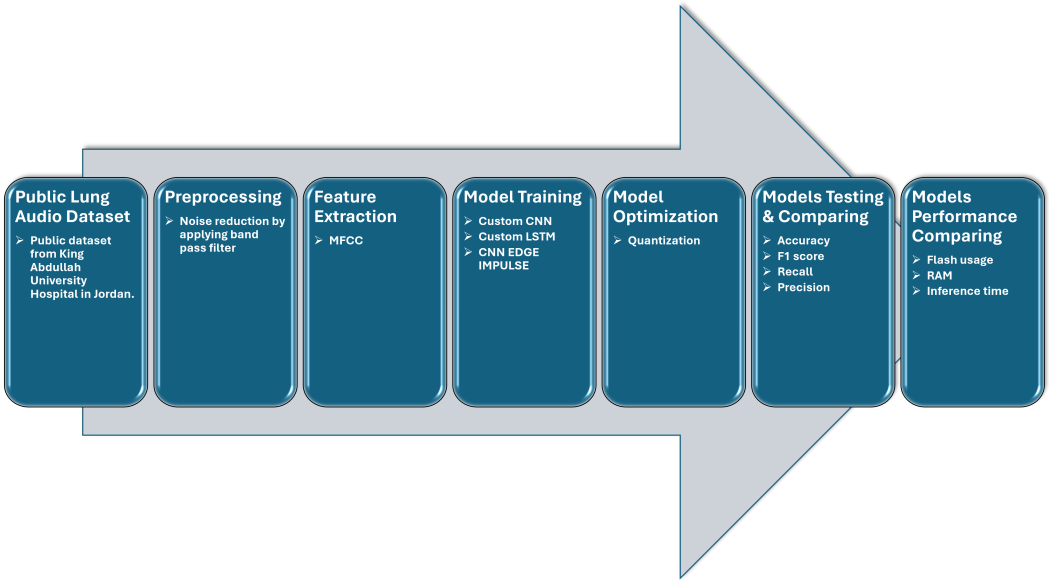

Figure 1. Workflow of lung sound analysis and model training process.

3.1. Dataset

In this study, we used a publicly available dataset from King Abdullah University Hospital in Jordan [37]. The dataset comprises 310 records from 105 patients with various respiratory conditions, including normal, asthma, pneumonia, heart failure, bronchiectasis, and chronic obstructive pulmonary disease. The duration of each recording ranges from 5 to 30 s, for a total of 88 min of data. The patients' ages range from 12 to 90 years, with a mean ± SD age of 48 ± 18 years. For each patient, three types of recordings were obtained, each using a different filtering mode—bell mode filtration, diaphragm mode filtration, and extended mode filtration—to minimize interference from heartbeats and external noise.

The audio recordings were captured by using a single channel at a sampling rate of 4 kHz. The audio files, in WAV format, were captured by using a single-channel stethoscope-based acquisition system (electronic stethoscope 3200; 3M Littmann) positioned at various locations on the chest wall. Table 2 presents the quantity of patients in each disease classification along with the associated count of recordings utilized from this dataset. The dataset demonstrates an unequal distribution, containing more data for the normal and asthma categories compared with the others. Given this situation, we decided to distinguish between two distinct kinds of lung sounds: normal and asthma. Before uploading data to

Edge Impulse, each audio file was processed with a 5th-order Butterworth bandpass filter, with upper and lower cut-off frequencies set to 100–1800 Hz, to reduce external noise [4,34].

Table 2. Demographic summary and recording details by disease category.

Category	Subjects	Age (Mean ± SD)	Number of Records	Duration
Normal	35 (11F, 24M)	43 ± 19	105	31 m 10 s
Asthma	32 (17F, 15M)	45 ± 15	94	28 m 06 s
Heart failure	21 (9F, 12M)	58 ± 18	56	13 m 59 s
Pneumonia	5 (2F, 3M)	55 ± 13	17	4 m 53 s
Bronchiectasis	3 (1F, 2M)	37 ± 26	9	2 m
COPD	9 (1F, 8M)	57 ± 9	29	07 m 54 s

Upon uploading our dataset, we proceeded to create an ML pipeline called an "impulse". This impulse comprises three primary building blocks: the input block, the processing block, and the learning block. The input block identifies the data type employed during the training of the model. It can either be an image or a time series. In our specific case, we utilized time series as the input data. The window size was set to 5000 ms, and the window increase was set to 5000 ms [34]. This duration is sufficient to cover at least one respiratory cycle, given the average resting respiration rates for adults (12–20 breaths per minute) [37].

3.2. Feature Extractor

Feature extraction is an essential phase in lung sound analysis, where raw audio signals are converted into useful representations for classification. This process includes several categories: time-domain, frequency-domain, and time–frequency-domain features. Time-domain features capture the temporal characteristics of lung sounds, including metrics such as the zero-crossing rate, root mean square, and signal envelope. Frequency-domain features provide insights into the energy distribution across different frequency bands and include measures like MFCCs. Time–frequency-domain features, like wavelet transform and spectrograms, present a combined view of both time and frequency characteristics [6].

MFCCs are commonly used as features [6,25,29] in lung analysis derived from the Fourier transform, which can capture the distribution of energy in different frequency bands. The MFCCs consist of a series of coefficients that capture the characteristics of the signal's spectrum. These coefficients are derived by taking the logarithm of the discrete cosine transform applied to the signal's spectrum. The particular parameters utilized to produce MFCCs are detailed in Table 3.

Table 3. Relevant parameters used to generate MFCCs.

Parameter	Value
Number of coefficients	13
Frame length	0.256
Frame stride	0.064
Filter number	20
FFT length	256

In order to obtain the MFCC parameters presented in Table 3, we were inspired by the methodology used in [34], where the authors tested multiple configurations of MFCC parameters to optimize model performance and resource consumption. Following a similar procedure, we conducted multiple iterative tests to explore different parameter settings and find the optimal balance between performance and computational efficiency. While we initially tested the parameters used in their study, we found that these configurations did

not yield the desired results in our specific case. As a result, we experimented with other configurations to achieve the best balance between model accuracy and resource efficiency.

Given the constraints of TinyML, where devices are resource-limited and power-sensitive, it was essential to choose parameters that minimized memory usage and computational time. The final selected parameters consume 38 KB of RAM and have a processing time of 763 ms, based on estimates provided by the Edge Impulse platform for an Arduino Nano 33 BLE [38].

3.3. Model Proposition

In this work, we developed three classifiers, i.e., a custom CNN model, the CNN model proposed by Edge Impulse, and a custom LSTM model, that can diagnose two distinct respiratory conditions, that is, asthma and normal. We drew inspiration from the existing literature [31,34], which demonstrated promising results in lung disease detection. Tables 4–6 illustrate the architectures of these three models.

Table 4. Our CNN model.

Custom CNN Model
Input(shape=(975,))
Reshape((975 / 13, 13))
Conv1D(32, kernel_size=3, activation='relu')
MaxPooling1D(pool_size=2)
BatchNormalization()
Dropout(20%)
Conv1D(64, kernel_size=3, activation='relu')
MaxPooling1D(pool_size=2)
BatchNormalization()
Dropout(20%)
Conv1D(128, kernel_size=3, activation='relu')
MaxPooling1D(pool_size=2)
BatchNormalization()
Dropout(20%)
Flatten()

Table 5. Custom LSTM model.

Our LSTM Model
Input(shape=(975,))
Reshape((975 / 13, 13))
LSTM(128)
Dropout(20%)
LSTM(64)
Dropout(20%)
LSTM(32)
Dropout(20%)
Dense(512, activation='relu')
Dropout(40)

Table 5. *Cont.*

Our LSTM Model
Dense(8, activation='relu')
Dropout(40%)
Dense(classes, activation='softmax')

Table 6. CNN model proposed by Edge Impule.

CNN Model Proposed by Edge Impule
Input(shape=(975,))
Reshape((975 / 13, 13))
Conv1D(8, kernel_size=3, activation='relu')
MaxPooling1D(pool_size=2)
Dropout(25%)
Conv1D(16, kernel_size=3, activation='relu')
MaxPooling1D(pool_size=2)
Dropout(25%)
Dense(classes, activation='softmax')

To tune the hyperparameters, a total of 396 combinations of epochs, learning rates, and mini-batch sizes were explored across all three models. Specifically, we tested epochs ranging from 10 to 300, with learning rates of 10^{-3}, 10^{-4}, 6×10^{-3}, and 6×10^{-4}, and mini-batch sizes of 32, 16, and 8. The Adam optimizer was consistently applied across all models to facilitate the optimization process. Table 7 summarizes the optimal hyperparameters, presenting the configurations that show the highest accuracy.

Table 7. Optimal hyperparameters showing the highest accuracy.

Model	**Learning Rate**	**Epochs**	**Batch Size**
Custom CNN	6×10^{-3}	300	32
CNN Edge Impulse	6×10^{-3}	100	16
Custom LSTM	6×10^{-3}	100	32

3.4. Target Device

The target device for deployment is an Arduino Nano 33 BLE Sense [38], featuring a Cortex-M4F 64 MHz processor, chosen for its compact size and versatility across various applications. With 256 kB of RAM and 1024 kB of ROM, the device has limited computational resources. To optimize model deployment, we used post-training quantization provided by the Edge Impulse platform, utilizing implementations from the TensorFlow Lite Micro library [39]. This technique reduces the precision of a model's internal representations by converting 32-bit floating-point parameters into lower-precision int8, significantly reducing the model's memory (ROM) requirements. This optimization makes deployment on a constrained device more feasible and accelerates computation, enabling quicker predictions and responses. Post-training quantization is thus a crucial step, preserving core functionality and accuracy while adapting the models to the device's limited resources.

Given the deployment on the Arduino Nano 33 BLE Sense, additional metrics, such as inference time, memory usage, and storage footprint, were considered. These factors are essential to ensuring the models' efficiency and smooth performance on the device. By evaluating classification metrics, computational efficiency, and suitability for the tiny

device, we made informed decisions regarding model selection. Edge Impulse offers a functionality that estimates the performance of the model on the target device before deployment, facilitating a more informed deployment process.

4. Results and Discussion

In this section, we compare the performance of the three models by using metrics such as accuracy, precision, recall, F1-score, and Area Under the Curve (AUC), as well as model size, inference time, and peak RAM usage, which are critical for deployment on TinyML devices. The dataset was split into 72% training, 10% validation, and 18% testing. Each model was trained and quantized by Edge Impulse for deployment suitability.

Table 8 shows the results of our experiments. the custom CNN model achieved the highest performance, with an accuracy of 96% on the test set and an AUC of 0.96. In contrast, the CNN Edge Impulse model, while faster and less resource-intensive, demonstrated lower accuracy, 85%, and an AUC of 0.85. This difference can be attributed to the simpler architecture of the Edge Impulse model. The custom LSTM model, however, achieved lower accuracy, 90%, compared with the CNN.

Table 8. Model evaluation on validation and testing sets.

Model	Accuracy	Precision	Recall	F1-Score	Loss Value	Area Under ROC Curve
Custom CNN						
Validation	100%	100%	100%	100%	0.03	1.00
Testing	96%	97%	97%	97%	-	0.96
CNN Edge Impulse						
Validation	92%	92%	92%	92%	0.24	0.91
Testing	85%	86%	86%	86%	-	0.85
Custom LSTM						
Validation	93%	93%	93%	93%	0.28	0.93
Testing	90%	93%	92%	92%	-	0.90

The confusion matrix in Table 9 provides a detailed evaluation of the classification performance of the CNN, LSTM, and CNN-EDGE-IMPULSE models on the test set data. The custom CNN outperformed the other models, correctly classifying 94.1% of "Asthma" cases and 98.3% of "Normal" cases, while the LSTM achieved 82.4% and 98.3%, respectively. CNN Edge Impulse, while more resource-efficient, had the lowest performance, with accuracy of 76.5% for "Asthma" and 88.1% for "Normal".

Table 9. Confusion matrix of test set data. Green: correct classifications, Red: misclassifications.

	Asthma (CNN)	Normal (CNN)	Asthma (LSTM)	Normal (LSTM)	Asthma (CNN-EDGE-IMPULSE)	Normal (CNN-EDGE-IMPULSE)	Uncertain
Asthma (CNN)	94.1%	5.9%	0%	0%	0%	0%	0%
Normal (CNN)	1.7%	98.3%	0%	0%	0%	0%	0%
Asthma (LSTM)	0%	0%	82.4%	17.6%	0%	0%	0%
Normal (LSTM)	0%	0%	1.7%	98.3%	0%	0%	0%
Asthma (CNN-EDGE-IMPULSE)	0%	0%	0%	0%	76.5%	20.6%	2.9%
Normal (CNN-EDGE-IMPULSE)	0%	0%	0%	0%	10.2%	88.1%	1.7%
F1-score	0.96	0.97	0.89	0.94	0.79	0.88	

Table 10 compares the models based on resource consumption, as estimated by the Edge Impulse cloud platform on the Arduino Nano 33. The CNN Edge Impulse model demonstrates its clear advantage for low-power, resource-constrained environments, requiring only 4.5 KB of RAM. However, this comes at the cost of lower classification performance. The custom CNN, which uses more RAM (12 KB) and has a longer inference time, strikes a balance between resource usage and accuracy, making it a more suitable option when both high classification performance and moderate resource usage are required.

Table 10. Performance comparison of different models.

Model	Inferencing Time	Peak RAM USAGE	Flash Usage
Our CNN	127 ms	12.0 K	249.6 K
CNN Edge Impulse	6 ms	4.5 K	31.7 K
Our LSTM	324 ms	23.2 K	190 K

On the other hand, the LSTM model, while providing good accuracy (90%) and the ability to process sequential data, has the highest resource demands, consuming 23.2 KB of RAM. This makes it less practical for highly resource-constrained devices.

Table 11 presents a comparative analysis between our proposed CNN model and similar works that utilize Edge ML models. Our TinyML model demonstrates superior performance, achieving higher accuracy, 96%, while also excelling in resource efficiency, making it more suitable for deployment on low-power devices.

Table 11. Qualitative performance comparison of the proposed CNN with existing works.

Work	[33]	[34]	Our Work
Dataset	KAUH	ICBHI 2017	**KAUH**
Feature	Mel-spectrogram	MFCC	**MFCC**
Preprocessing	Discrete Fourier transform (DFT)-based filtering and segmentation into 5 s windows	Zero padding, segmentation into 3 s windows, Butterworth bandpass filter, and Z-score normalization	**Zero padding, Butterworth bandpass filter, and segmentation into 5 s windows**
Lung diseases	Asthma and healthy	Normal and subnormal	**Asthma and healthy**
Target device	-	Alinx AX7A200 FPGA	**Arduino Nano 33 BLE**
Model	RDsLINet	LSTM	**CNN**
Accuracy	91%	81.4%	**96%**
Total execution time	5.439 s	-	**890 ms**
Peak RAM USAGE	-	32 KB	**12.0 KB**
Flash usage	498 KB	-	**249.6 KB**

One of the key reasons our model outperforms that of [33], which used the same dataset, lies in the application of pruning and quantization techniques. These methods allowed us to significantly reduce both model size and inference time, optimizing the model for resource-constrained environments. Pruning effectively removes less critical weights from the network, thus speeding up computation and reducing memory usage, while quantization lowers the precision of the model's parameters without substantially affecting its accuracy, leading to more efficient deployment on embedded devices.

In contrast, in [33], the authors applied depthwise separable convolutions and global average pooling (GAP) layers instead of fully connected layers to reduce the model size and execution time. This likely contributes to the differences in execution time and accuracy observed in our model compared with [33].

5. Conclusions and Future Work

In this work, we have exploited TinyML models for detecting respiratory diseases, particularly asthma, using lung sound recordings. Our custom CNN model achieved an accuracy of 96% while maintaining efficient resource usage. This demonstrates the feasibility of deploying real-time, accurate diagnostic tools on resource-constrained devices, making them suitable for portable medical applications.

The potential impact of this approach on healthcare is significant. By offering a low-cost, portable solution for respiratory disease detection, our models can enhance access to reliable diagnostics in remote and underserved areas, reducing the reliance on traditional medical facilities and expensive equipment. This advancement is crucial to improving early disease detection and patient outcomes.

In future work, we will address the challenges encountered with dataset imbalance, which limited the diversity of the training data. To overcome this challenge, we will explore merging multiple publicly available lung sound datasets and applying data augmentation techniques, such as variational autoencoders [30]. Additionally, real-world clinical validation and the inclusion of more respiratory conditions will be key steps toward refining and extending the applicability of our models.

Author Contributions: Conceptualization, Y.A.; Methodology, Y.A., N.B. and H.C.; Validation, Y.A.; Formal analysis, Y.A.; Investigation, Y.A.; Data curation, Y.A.; Writing—original draft, Y.A.; Writing—review & editing, N.B. and M.B.; Supervision, N.B. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are publicly available at https://data.mendeley.com/datasets/jwyy9np4gv/3 (accessed on 21 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. World Health Organization. The Top 10 Causes of Death. 2021. Available online: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death (accessed on 31 March 2024).
2. Hashoul, D.; Haick, H. Sensors for detecting pulmonary diseases from exhaled breath. *Eur. Respir. Rev.* **2019**, *28*. [CrossRef] [PubMed]
3. Sfayyih, A.H.; Sulaiman, N.; Sabry, A.H. A review on lung disease recognition by acoustic signal analysis with deep learning networks. *J. Big Data* **2023**, *10*, 101. [CrossRef] [PubMed]
4. Andrès, E.; Gass, R.; Charloux, A.; Brandt, C.; Hentzler, A. Respiratory sound analysis in the era of evidence-based medicine and the world of medicine 2.0. *J. Med. Life* **2018**, *11*, 89. [PubMed]
5. Alqudah, A.M.; Qazan, S.; Obeidat, Y.M. Deep learning models for detecting respiratory pathologies from raw lung auscultation sounds. *Soft Comput.* **2022**, *26*, 13405–13429. [CrossRef] [PubMed]
6. Huang, D.M.; Huang, J.; Qiao, K.; Zhong, N.S.; Lu, H.Z.; Wang, W.J. Deep learning-based lung sound analysis for intelligent stethoscope. *Mil. Med. Res.* **2023**, *10*, 44. [CrossRef]
7. McLane, I.; Emmanouilidou, D.; West, J.E.; Elhilali, M. Design and Comparative Performance of a Robust Lung Auscultation System for Noisy Clinical Settings. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 2583–2594. [CrossRef]
8. Seah, J.J.; Zhao, J.; Wang, D.Y.; Lee, H.P. Review on the advancements of stethoscope types in chest auscultation. *Diagnostics* **2023**, *13*, 1545. [CrossRef]
9. Lella, K.K.; Jagadeesh, M.; Alphonse, P. Artificial intelligence-based framework to identify the abnormalities in the COVID-19 disease and other common respiratory diseases from digital stethoscope data using deep CNN. *Health Inf. Sci. Syst.* **2024**, *12*, 22. [CrossRef]
10. Tsoukas, V.; Boumpa, E.; Giannakas, G.; Kakarountas, A. A review of machine learning and tinyml in healthcare. In Proceedings of the 25th Pan-Hellenic Conference on Informatics, Volos, Greece, 26–28 November 2021; pp. 69–73. [CrossRef]

11. Abadade, Y.; Temouden, A.; Bamoumen, H.; Benamar, N.; Chtouki, Y.; Hafid, A.S. A Comprehensive Survey on TinyML. *IEEE Access* **2023**, *11*, 96892–96922. [CrossRef]
12. Ooko, S.O.; Muyonga Ogore, M.; Nsenga, J.; Zennaro, M. TinyML in Africa: Opportunities and Challenges. In Proceedings of the 2021 IEEE Globecom Workshops (GC Wkshps), Madrid, Spain, 7–11 December 2021; pp. 1–6. [CrossRef]
13. Ray, P.P. A review on TinyML: State-of-the-art and prospects. *J. King Saud Univ. Comput. Inf. Sci.* **2022**, *34*, 1595–1623. [CrossRef]
14. Dutta, D.L.; Bharali, S. TinyML Meets IoT: A Comprehensive Survey. *Internet Things* **2021**, *16*, 100461. [CrossRef]
15. Nicolas, C.; Naila, B.; Amar, R.C. TinyML Smart Sensor for Energy Saving in Internet of Things Precision Agriculture platform. In Proceedings of the 2022 Thirteenth International Conference on Ubiquitous and Future Networks (ICUFN), Barcelona, Spain, 5–8 July 2022; pp. 256–259. [CrossRef]
16. Bhamare, M.; Kulkarni, P.V.; Rane, R.; Bobde, S.; Patankar, R. Chapter 14—TinyML applications and use cases for healthcare. In *TinyML for Edge Intelligence in IoT and LPWAN Networks*; Academic Press: Cambridge, MA, USA, 2024; pp. 331–353. [CrossRef]
17. Bamoumen, H.; Temouden, A.; Benamar, N.; Chtouki, Y. How TinyML Can be Leveraged to Solve Environmental Problems: A Survey. In Proceedings of the 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 20–21 November 2022; pp. 338–343. [CrossRef]
18. Diab, M.S.; Rodriguez-Villegas, E. Embedded Machine Learning Using Microcontrollers in Wearable and Ambulatory Systems for Health and Care Applications: A Review. *IEEE Access* **2022**, *10*, 98450–98474. [CrossRef]
19. Sun, B.; Bayes, S.; Abotaleb, A.M.; Hassan, M. The Case for tinyML in Healthcare: CNNs for Real-Time On-Edge Blood Pressure Estimation. In Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing, Tallinn, Estonia, 27–31 March 2023; pp. 629–638. [CrossRef]
20. Saadeh, W.; Butt, S.A.; Altaf, M.A.B. A Patient-Specific Single Sensor IoT-Based Wearable Fall Prediction and Detection System. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 995–1003. [CrossRef]
21. Fang, K.; Xu, Z.; Li, Y.; Pan, J. A Fall Detection using Sound Technology Based on TinyML. In Proceedings of the 2021 11th International Conference on Information Technology in Medicine and Education (ITME), Wuyishan, China, 19–21 November 2021; pp. 222–225. [CrossRef]
22. Zhu, T.; Kuang, L.; Li, K.; Zeng, J.; Herrero, P.; Georgiou, P. Blood Glucose Prediction in Type 1 Diabetes Using Deep Learning on the Edge. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May 2021. [CrossRef]
23. Risso, M.; Burrello, A.; Pagliari, D.J.; Benatti, S.; Macii, E.; Benini, L.; Pontino, M. Robust and Energy-Efficient PPG-Based Heart-Rate Monitoring. In Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), Daegu, Republic of Korea, 22–28 May 2021. [CrossRef]
24. Alghamdi, N.S.; Zakariah, M.; Karamti, H. A deep CNN-based acoustic model for the identification of lung diseases utilizing extracted MFCC features from respiratory sounds. In *Multimedia Tools and Applications*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–33. [CrossRef]
25. Ullah, A.; Khan, M.S.; Khan, M.U.; Mujahid, F. Automatic Classification of Lung Sounds Using Machine Learning Algorithms. In Proceedings of the 2021 International Conference on Frontiers of Information Technology (FIT), Islamabad, Pakistan, 13–14 December 2021; pp. 131–136. [CrossRef]
26. Abdul, Z.K.; Al-Talabani, A.K. Mel Frequency Cepstral Coefficient and its Applications: A Review. *IEEE Access* **2022**, *10*, 122136–122158. [CrossRef]
27. Owens, F.; Murphy, M. A short-time Fourier transform. *Signal Process.* **1988**, *14*, 3–10. [CrossRef]
28. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef]
29. Sreeram, A.; Ravishankar, U.; Sripada, N.R.; Mamidgi, B. Investigating the potential of MFCC features in classifying respiratory diseases. In Proceedings of the 2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Paris, France, 14–16 December 2020; pp. 1–7. [CrossRef]
30. García-Ordás, M.T.; Benítez-Andrades, J.A.; García-Rodríguez, I.; Benavides, C.; Alaiz-Moretón, H. Detecting Respiratory Pathologies Using Convolutional Neural Networks and Variational Autoencoders for Unbalancing Data. *Sensors* **2020**, *20*, 1214. [CrossRef]
31. Aykanat, M.; Kılıç, Ö.; Kurt, B.; Saryal, S. Classification of lung sounds using convolutional neural networks. *EURASIP J. Image Video Process.* **2017**, *2017*, 65. [CrossRef]
32. Tawfik, M.; Al-Zidi, N.M.; Fathail, I.; Nimbhore, S. Asthma Detection System: Machine and Deep Learning-Based Techniques. In *Artificial Intelligence and Sustainable Computing;* Pandit, M., Gaur, M.K., Rana, P.S., Tiwari, A., Eds.; Springer: Singapore, 2022; pp. 207–218. [CrossRef]
33. Roy, A.; Satija, U. RDLINet: A Novel Lightweight Inception Network for Respiratory Disease Classification Using Lung Sounds. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 4008813. [CrossRef]
34. Zhou, W.; Yu, L.; Zhang, M.; Xiao, W. A low power respiratory sound diagnosis processing unit based on LSTM for wearable health monitoring. *Biomed. Eng. Tech.* **2023**, *68*, 469–480. [CrossRef]
35. Harvard. AI for Good-Healthcare. 2024. Available online: https://harvard-edge.github.io/cs249r_book/contents/ai_for_good/ai_for_good.html#healthcare (accessed on 31 March 2024).

36. Hymel, S.; Banbury, C.; Situnayake, D.; Elium, A.; Ward, C.; Kelcey, M.; Baaijens, M.; Majchrzycki, M.; Plunkett, J.; Tischler, D.; et al. Edge Impulse: An MLOps Platform for Tiny Machine Learning. *arXiv* **2023**, arXiv:2212.03332.
37. Fraiwan, M.; Fraiwan, L.; Khassawneh, B.; Ibnian, A. A dataset of lung sounds recorded from the chest wall using an electronic stethoscope. *Data Brief* **2021**, *35*, 106913. [CrossRef] [PubMed]
38. Arduino. Nano 33 BLE Sense. Available online: https://docs.arduino.cc/hardware/nano-33-ble-sense/ (accessed on 13 October 2024).
39. David, R.; Duke, J.; Jain, A.; Janapa Reddi, V.; Jeffries, N.; Li, J.; Kreeger, N.; Nappier, I.; Natraj, M.; Wang, T.; et al. TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems. *Proc. Mach. Learn. Syst.* **2021**, *3*, 800–811.