engineering proceedings

MDPI

Proceeding Paper

A TinyML Approach to Real-Time Snoring Detection in Resource-Constrained Wearables Devices †

Timothy Malche [1], Sumegh Tharewal [2] and Priti Maheshwary [3]

[1] Department of Computer Applications, Manipal University Jaipur, Jaipur, Rajasthan, India; timothy.malche@jaipur.manipal.edu
[2] School of Advance Computing, DBS Global University, Dehradun, Uttarakhand, India, sumeghtharewal@gmail.com
[3] Department of Computer Science & Engineering, Rabindranath Tagore University, Bhopal, Madhya Pradesh, India; pritimaheshwary@gmail.com
* Correspondence: timothy.malche@jaipur.manipal.edu
† Presented at The 11th International Electronic Conference on Sensors and Applications (ECSA-11), 26–28 November 2024; Available online: https://sciforum.net/event/ecsa-11.

Abstract: This study proposes a health monitoring system for snoring detection utilizing Tiny Machine Learning (TinyML) models, specifically designed for resource-constrained wearable Internet of Things (IoT) devices. This research addresses significant constraints associated with running Machine Learning models on IoT devices, such as latency, limited memory, and low computational resources. These parameters are essential for real-time monitoring in healthcare applications, where prompt response is critical. The research focuses on developing a TinyML model capable of identifying specific audio patterns related to snoring during sleep. Experimental evaluations conducted in real-world sleep environments with the TinyML model deployed on resource-constrained wearable IoT devices. The evaluation results show that the proposed model achieves high accuracy while utilizing minimal computational resources and without introducing latency issues. The integration of Audio (Syntiant) and advanced audio preprocessing techniques, the proposed system improves the efficiency of the TinyML model on wearable devices. The quantized TinyML model achieved accuracy of 95.85% with a low latency of 48 ms, utilizing only 17.0 K RAM and 34.07 K flash memory for real-time snoring classification. This study highlights the benefits of practical deployment of TinyML model for snoring detection on resource-constrained wearable IoT devices, demonstrating that such models can operate effectively within the constraints of current wearable technology.

Keywords: Wearable Sensors; Healthcare Monitoring; Internet of Things; TinyML; Edge AI

Citation: Malche, T.; Tharewal, S.; Maheshwary, P. A TinyML Approach to Real-Time Snoring Detection in Resource-Constrained Wearables Devices. *Eng. Proc.* **2024**, *6*, x. https://doi.org/10.3390/xxxxx

Academic Editor(s): Name

Published: 26 November 2024

1. Introduction

The studies conducted on human lives reveals that close relationships are very important for sustaining happiness more than money or anything else [1]. Snoring not only disrupts the snorer's sleep but can also foster bitterness and resentment between couples. Studies have also shown that almost 30–40% humans are habitual to snoring [2]. When muscles around throat relax during sleep, it narrows the airway which results in vibration due to which snoring happen [3]. Snoring often related to symptom of a sleep disorder. It emphasizes the importance of detecting and addressing snoring to enhance persons' quality of life. This study aims to build a TinyML (Tiny Machine Learning)-based device dedicated to the detect and alert when a person is snoring [4].

Building a snoring detection device has some challenges, ranging from technical hurdles to user-related considerations. The snoring patterns can also vary from person to person. Therefore, designing a device that can accurately captures and interprets different snoring voices is a big challenge. Environmental noise, such as ambient noise in the

bedroom or external disturbances can also impact accurate snore detection. To ensure the the snoring device is comfortable for users to wear during sleep is also a main challenge. Building a machine learning model for resource-constrained devices, such as wearable devices, is also a major issue. The size and complexity of machine learning models can also directly impact the storage limitation of such wearable devices. Achieving real-time inference on resource-constrained devices is also challenging audio signals has to be processed for snoring detection. On the other hand, TinyML for building a wearable device for snore detection offers many benefits as follows:

- TinyML models are specifically designed for resource-constrained environments.
- TinyML models are optimized for real-time inference. It enables on-device inference and eliminates the need for continuous data transmission to external servers for analysis.
- TinyML models ensures real-time response because of low latency. In the context of snore detection, this capability is critical for providing timely alerts.
- TinyML models are characterized by their compact size, making it suitable for deployment in wearable devices.

2. Related Work

The study in [5] presents a novel snore detection algorithm which uses convolutional recurrent neural networks. The model is evaluated on audio data of from 38 users while they are sleeping. The system uses microphone installed at different places and the algorithm achieved high accuracy (95.3%) in detecting snore events, with 92.2% sensitivity and 97.7% specificity. The performance of algorithm remained robust across different microphone positions which indicate the reliability of the system for snore detection in different sleep environments. Another study in [6] focuses on the crucial need for a reliable snoring detection system for monitoring and diagnosing obstructive sleep apnea (OSA) to improve the quality of life for those with the disorder. This research proposes a hybrid convolutional neural network (CNN) model for detecting snores. For OSA monitoring in real-world situations, the model achieved an 89.3% average classification accuracy, 89.7% sensitivity, and 88.5% specificity. The research in [7] discusses the health hazards associated with obstructive sleep apnea hypopnea syndrome (OSAHS) and suggests a novel strategy for monitoring and identification to avoid treatment delays. The system classifies snoring voices of normal individuals and those with OSAHS. Mel-frequency cepstral coefficients (MFCC) are used in the study, along with CNN and LSTM models for feature extraction. The method has the greatest accuracy rate of 87% for binary categorization of snoring data. The suggested approach can also estimate the severity of OSAHS using the determined AHI value, providing useful information for clinical diagnosis and therapy. In [8] authors discuss the challenges of detecting OSAHS by developing a snore detection system that enables at-home screening. The study suggests an approach for utilizing the hardware limitation of smartphones. The developed system detects snoring and environmental noises with using real-time snore detector (RTSD) for sleep sound recordings. The RTSD serves as a valuable standalone tool for analyzing quality of sleep.

The research in [9] explores about snoring issue and highlight its impact on health. The research aimed to develop a deep learning model implemented as an Android smartphone app for snore detection. The app analyzes real-time audio captured by the on-device microphone and classifies snore and non-snore voice and noise with an accuracy of 98%. Authors in [10] discussed an efficient method for snoring detection to diagnose OSA and health complications related to it. The study employs a CNN to distinguish between snoring and non-snoring voices and noises based on audio inputs. The raw data is preprocessed using MFCC, and multi-scale features are extracted from the frequency domain using a multi-branch CNN (MBCNN). The developed model achieves a snoring detection accuracy of 99.5%. The research in [11] addresses the issue of poor sleep quality in modern society and proposed a method to detect snoring and coughing episodes

during sleep. The study discuss three stage method consisting of segmentation of nightly sound data into individual events, extraction of features from snoring and coughing episodes using Fourier Transform, and recognition of these events using Support Vector Machine (SVM) and Hidden Markov Model (HMM). Experimental results demonstrate the effectiveness of the method in accurately detecting snoring and coughing events. The study presented in [12] proposes a low-cost alternative to polysomnography (PSG) for detecting OSA. A repository of snoring audio recordings is processed using multi-threshold endpoint detection and feature extraction to obtain distinctive information. Machine learning models are then trained to predict feature categories. A real-time system for an embedded device is developed to detect snoring and OSA. A multi-classification temporal convolutional network (TCN) is trained to distinguish between non-snoring, snoring noises, and OSA-related snoring. The model achieved a high detection accuracy of 96.7% for OSA-related snoring.

Based on the analysis of the research review, it is concluded that there is a need for the development of an efficient, real-time wearable device and system capable of accurately detecting snoring. The system should address the limitations of resource-constrained devices and eliminate the dependency on cloud servers for inferencing. By creating a wearable device that can accurately detect snoring in real-time without relying on external servers, the people with OSA can benefit from continuous monitoring and timely intervention.

3. Methods

3.1. System Design

To detect snoring when a person is sleeping, a wearable device is designed using Arduino Nicla Voice [13]. The device contains built in microphone to receive snoring sound and provide input to TinyML model that was built. The main features of the device are given in Table 1:

Table 1. Device Configuration.

Microprocessor	Syntiant® NDP120 Neural Decision Processor™ (NDP)
Microcontroller	nRF52832 64 MHz (Arm Cortex M4)
Sensor	Microphone IM69D130
Power	3.7 V Li-po battery
Memory	512 KB Flash, 64 KB SRAM 16 MB SPI Flash, 48 KB SRAM
Connectivity	Bluetooth® Low Energy (ANNA-B112)

The device is powered by 3.7 V Li-po battery as illustrated in Figure 1.

As shown in Figure 1, the device can be comfortable worn in the wrist by the user during sleep and is used to detect snore and send alert to user. The System architecture is shown in Figure 2.

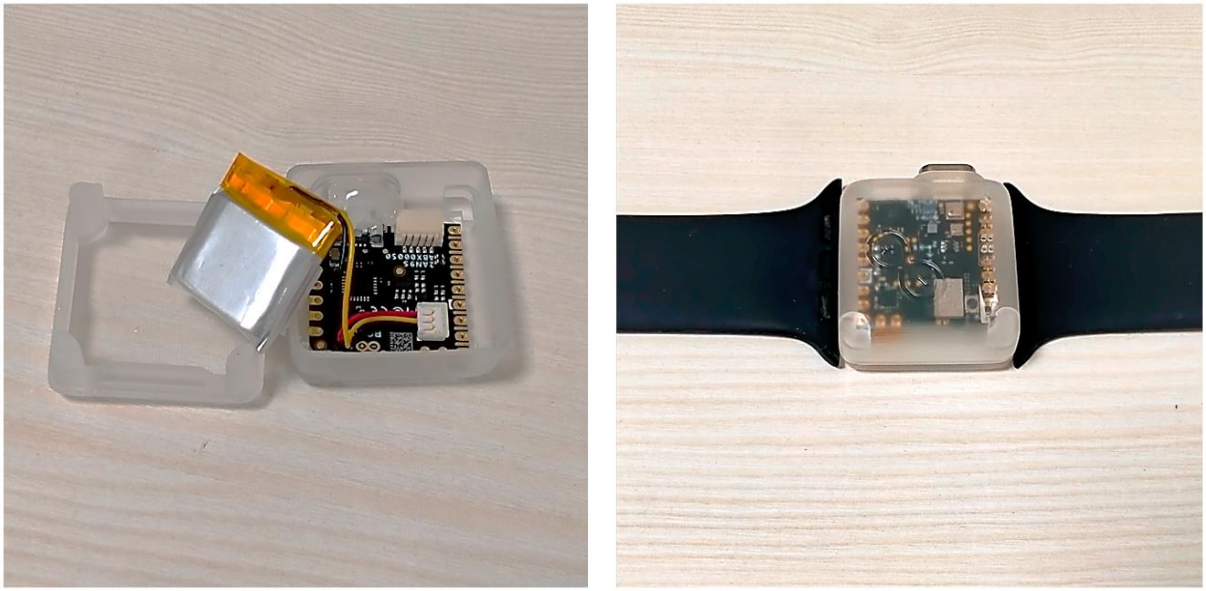

Figure 1. Wearable Sensor Design.

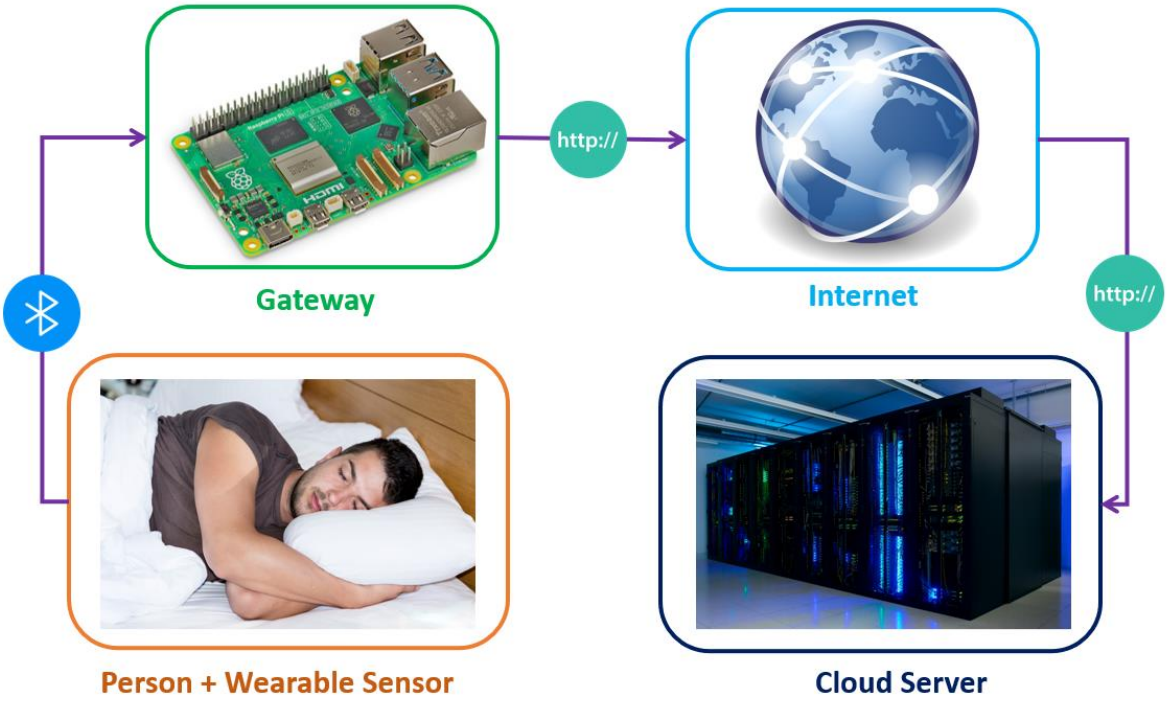

Figure 2. System Architecture.

The system works by detecting the snore sound using wearable sensor. If the snoring is detected by TinyML model running on wearable device, it sends the signal to nearby gateway device which in turn generate an alarm. The gateway also stores the snore detection data with timestamp locally and further sends it to cloud server for permanent storage and analysis. In this way the system not only detect and alert users for snoring but also keeps track of the historical data which may be shared to medical practitioners for further analysis and diagnosis.

3.2. Dataset

The dataset consists of two distinct classes that are snoring and non-snoring. The snoring class has 1000 samples of snoring voices, each lasting for 1 s. This collection includes snoring voices of adult men and women, children. The data set contains snoring and non-snoring voices and noises with and without background.

The non-snoring class consists of 500 samples of non-snoring voices and noises, each also lasting for 1 s. These samples contain varied background sounds. The dataset consists of 50 samples from each category of non-snoring voices and sounds. The dataset used for non-snoring sound consists of audio recordings from various categories, including door opening and closing, silence with motor vibrations, clock ticking, toilet flushing, vehicle sirens, rain and thunderstorms, streetcar sounds, human speech, and television news. These datasets were obtained from Kaggle [14]. Figure 3 provides a visual representation of the dataset used.

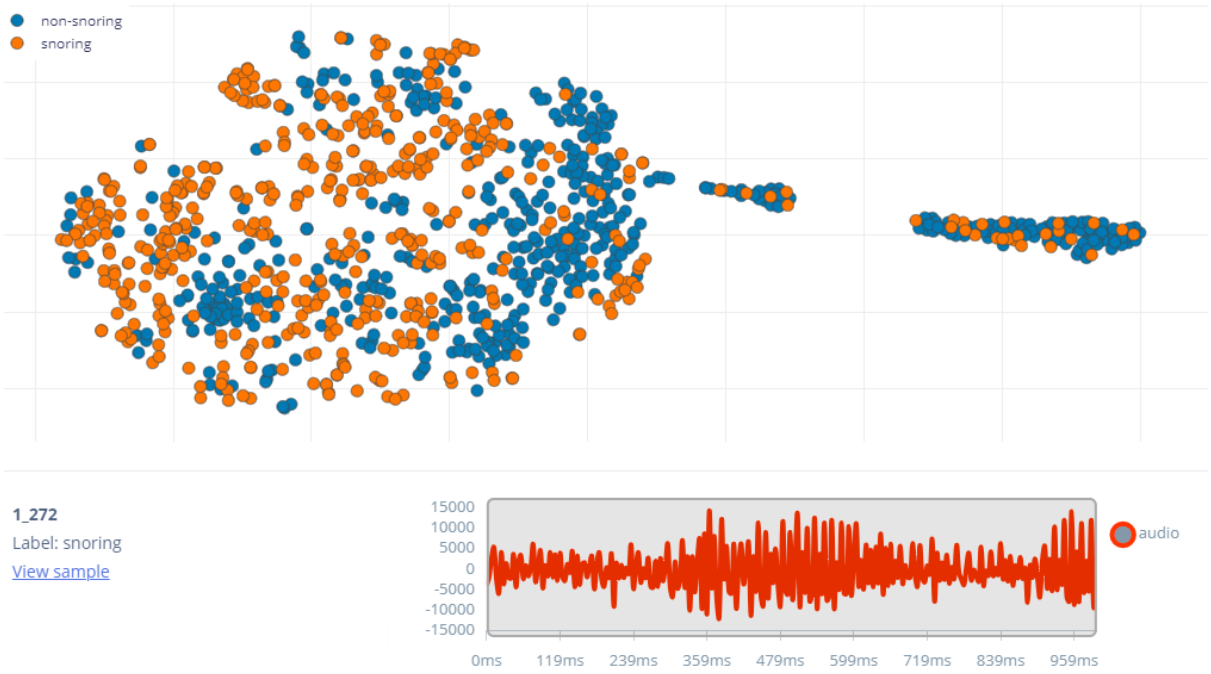

Figure 3. Dataset for snoring and non-snoring sounds.

3.3. Processing

In this study, Audio Syntiant [15] processing is utilised to classify snoring from non-snoring voices and noises. The Audio Syntiant is used to extract time and frequency information from signals. The Syntiant audio processing pipeline is a specialized version of Audio MFE, with additional steps for the Syntiant chip's unique characteristics. Key parameters include frame length, frame stride, filter number, FFT length, low frequency, and high frequency, which define the spectrogram features extracted using Mel-filterbank energy. Pre-emphasis is applied with a specified coefficient. The chip-specific features extractor is chosen based on the particular Syntiant chip used, ensuring optimal feature generation for the given hardware.

Syntiant's feature extraction process begins with a pre-emphasis step to amplify high-frequency components. The audio signal is divided into overlapping segments, with the frame length and stride determining the size and spacing of these segments. These parameters influence the extracted speech features. Table 2 provides the specific values used for Audio Syntiant.

Table 2. Audio Syntiant Parameters.

Log Mel filterbank energy features	
Frame length:	0.032
Frame stride:	0.024
Filter number:	40
FFT length:	512
Low frequency:	0
High frequency:	0
Preemphasis	
Coefficient:	0.96875
Chip	
Features extractor:	log-bin (NDP1 20/200)

The Figure 4 shows the DSP results as Syntiant spectrogram of the Snoring sound and Figure 5 visualizes the features generated for snoring and non-snoring voices and noises.

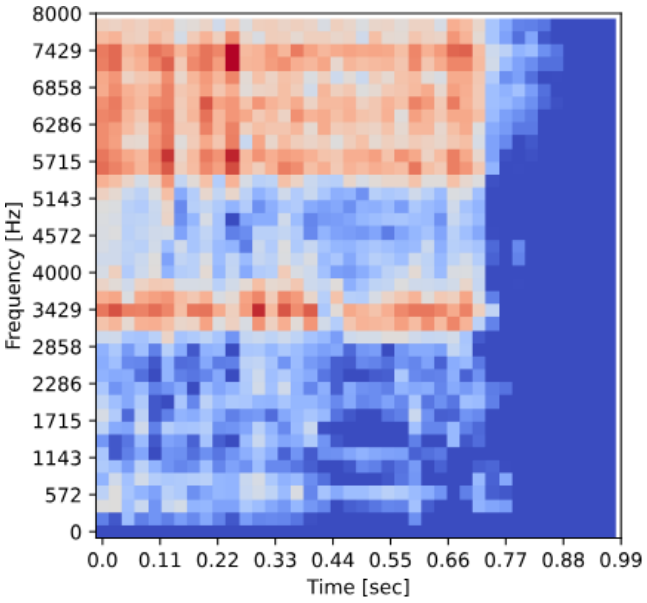

Figure 4. Syntiant spectrogram of the Snoring sound.

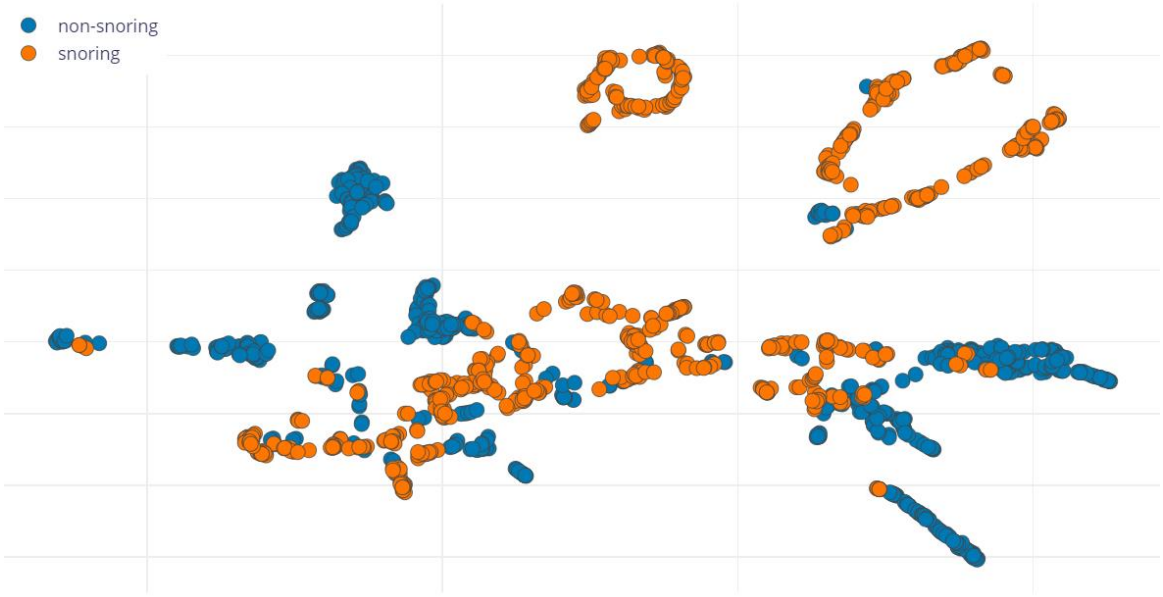

Figure 5. Generated Features for non-snoring and snoring sounds.

3.4. Model Architecture

The neural network architecture, as shown in Figure 6, is built using sequential model. The input to the network consists of a 2D array of 40 × 40 × 1 represents MFCC features. The first layer is 2D convolutional layer which has 8 filters of size 3 × 3. The filters are constrained using max-norm regularization with a maximum norm of 1. The ReLU activation function is used in this layer. After this a max pooling layer with a pool size 2 × 2 and a stride 2 is applied. Another 2D convolutional layer having 8 filters of 3 × 3 size is applied. A 'valid' padding, max-norm constraint, and ReLU activation is added to capture more complex features in this layer. Another similar max pooling layer follows the second convolutional layer. A dropout layer with a rate of 0.25 is included after the pooling layers to prevent overfitting. An average pooling layer with a dynamically determined pool size is used to aggregate features spatially before flattening. A reshape layer converts the 2D pooled feature maps into a 1D vector, preparing them for the fully connected layers. A fully connected layer having 16 neurons and ReLU activation is used, with L1 regularization to encourage sparsity in the learned weights. A second fully connected layer with 8 neurons and ReLU activation is added, also with L1 regularization and a dropout layer. The final fully connected layer has 2 neurons, representing the number of classes, with a softmax activation function to provide class probabilities.

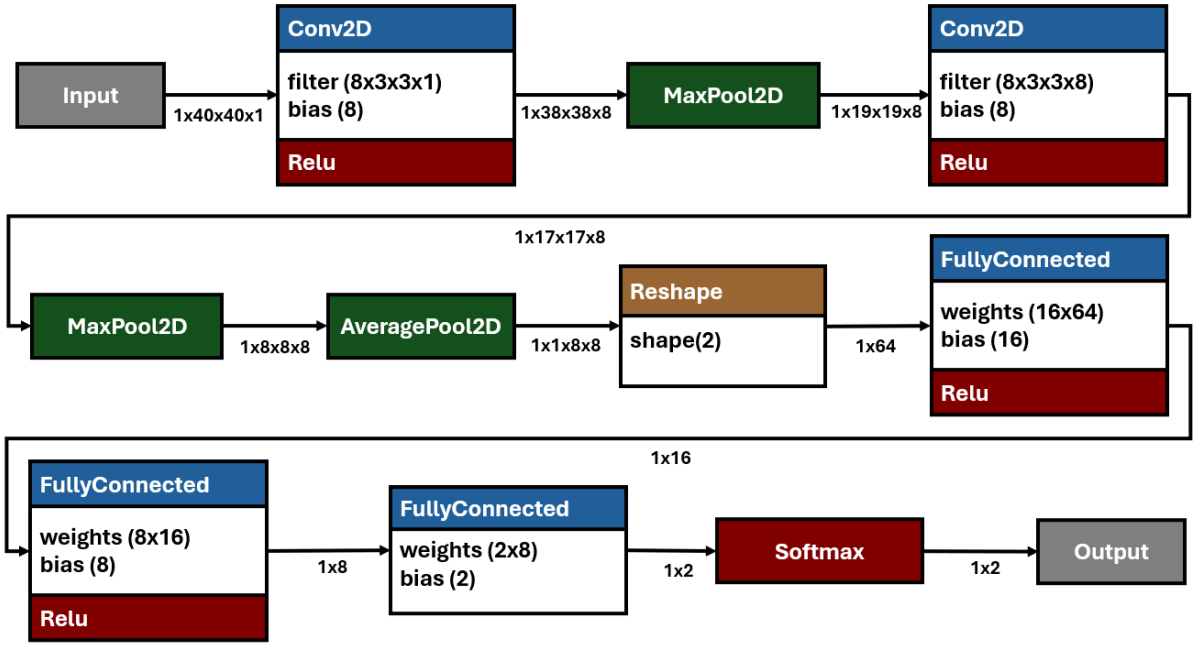

Figure 6. Model Architecture.

This architecture is designed to meet the specific requirements of the snore detection problem. For training the model, the 100 epochs, 0.0005 learning rate and 32 batch size was used. After training the model achieved the total accuracy of 96.6% with 0.11 loss. Finally, the TinyML model is quantized from float32 to int8 to fit in the requirements of low-resource device making it work efficiently on wearable devices. The Table 3 show confusion matrix for training dataset.

Table 3. Confusion Matrix (Training Dataset).

	Non-Snoring	**Snoring**
Non-Snoring	98.8%	1.2%
Snoring	5.7%	94.3%
F1 Scores	0.97	0.96

The graph in Figure 7 shows that model has accurately identified snoring and non-snoring voices and noises during training. The model architecture is shown in Figure 7.

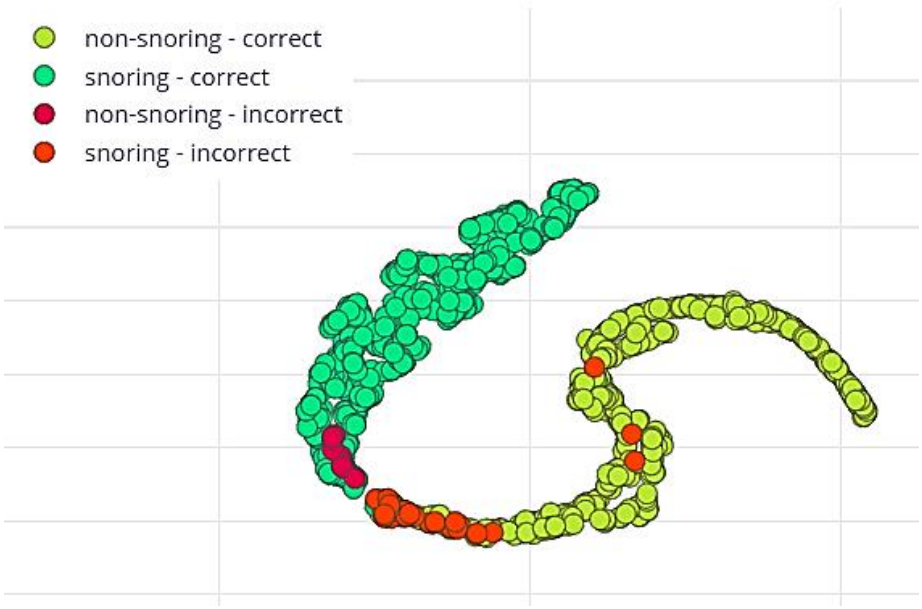

Figure 7. Model Accuracy during Training for snoring 94.3% and non-snoring 98.8%.

4. Results and Discussion

The trained model is evaluated on the test dataset. The accuracy of the model as well as the memory it takes, and its processing speed are also tested on resource constrained IoT devices. The confusion matrix for test dataset is shown in Table 4. For non-snoring predictions, the model correctly classified 96.8% of non-snoring instances as non-snoring and incorrectly classified 1.1% of non-snoring instances as snoring. It has also classified 2.1% of non-snoring instances as uncertain indicating tht model could not confidently classify the instance as either snoring or non-snoring. Similarly, the model correctly classified 96.4% of snoring instances as snoring and incorrectly classified 0.5% of snoring instances as non-snoring. It also classified 3.1% of snoring instances as uncertain. F1 Scores are measures of accuracy of the model and the F1 scores for "Non-Snoring" and "Snoring" voices are 0.98, which indicate that the model achieved high accuracy in classifying both classes on new data. The graph in Figure 8 show the model performance on test dataset.

Table 4. Confusion Matrix (Test Dataset).

	Non-Snoring	**Snoring**	**Uncertain**
Non-Snoring	96.8%	1.1%	2.1%
Snoring	0.5%	96.4%	3.1%
F1 Scores	0.98	0.98	

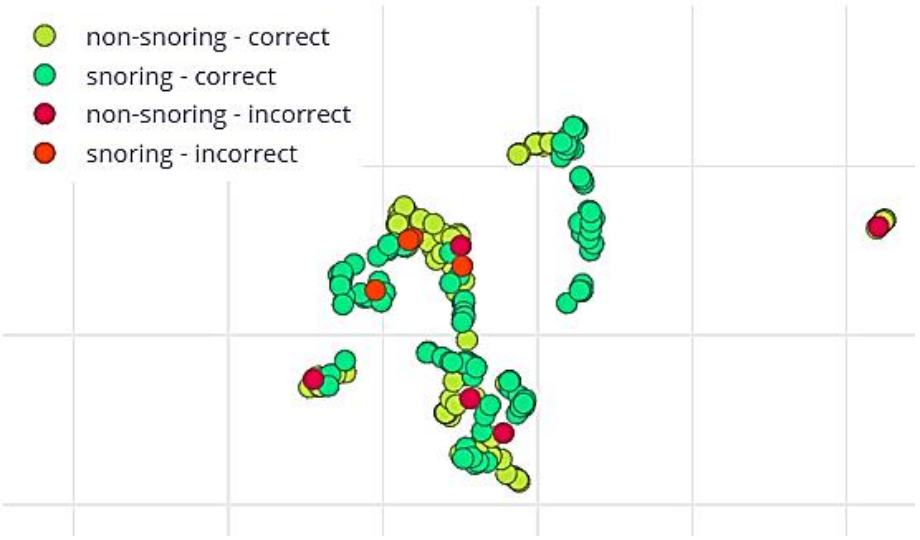

Figure 8. Model performance on Test Dataset.

The main objective of this research is to find the suitable model architecture that can be deployed in resource constrained environment of the target wearable device. This requires analysing the hardware requirements of the target device and accordingly designing and selecting the bet TinyML model to achieve maximum performance and accuracy. The analysis on input data, signal processing, and neural network structures have been conducted to build efficient model architecture as per the need of computing power and memory requirements of the device. The following architecture and configurations were used to deploy a TinyML model:

- Dataset Category: voice events
- Target Device: Cortex M4
- Time per inference: 100 ms
- Target RAM: 340 KB
- Target ROM: 1024 KB

After a thorough investigation the best model for the device is selected and the model is quantized to be deployed in the target device. The following Table 5 provides the comparison of the converted TensorFlow Lite model (float32) and the quantized model (int8) in terms of latency, memory and accuracy.

Table 5. Model Comparision.

	Unoptimized Model (FLOAT32)		**Quantized Model (INT8)**	
	Classifier	**Total**	**Classifier**	**Total**
Latency	955 ms	955 ms	48 ms	48 ms
RAM	58.6 K	58.6 K	17.0 K	17.0 K
Flash	32.8 K		34.7 K	
Accuracy	**96.63%**		**95.85%**	

5. Conclusions

This study proposes a health monitoring system specifically developed for snoring detection using Tiny Machine Learning (TinyML) models. The primary objective of the research is to develop a TinyML-based snoring detection model to accurately identify specific audio patterns associated with snoring during sleep. The research also aimed at designing TinyML model for resource-constrained Internet of Things (IoT) devices. To test the efficacy of the proposed model, the experiments are conducted in real-world sleep environments by deploying the TinyML model to resource constrained wearable IoT

device. The results have shown that model has achieved high accuracy and while using the minimal computational resource on the target wearable device. The quantized model achieved accuracy of 95.85% with the low latency of 48 milliseconds, RAM 17.0 K and Flash 34.7 K making the model ideal choice for wearable devices. This research demonstrates the feasibility and practicality of implementing snoring detection TinyML models on resource constrained IoT devices and provides a non-intrusive method of sleep monitoring. The proposed research makes an important contribution to the advancement of TinyML applications in health monitoring.

Author Contributions: T. M. designed methodology and build the TinyML model; S. T. conceptualized the idea for this manuscript, supervised and administered the work, prepared the original draft, validated the data and results; P. M. designed hardware, validated the data and results.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable

Informed Consent Statement: Not Applicable

Data Availability Statement: The data presented in this study are available on request from the first author.

Acknowledgments: The TinyML model and the graphs for this study was generated using Edge-impulse [16] platform.

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

References

1. Mineo, L. Good Genes Are Nice, But Joy Is Better. The Harvard Gazette. Available online: https://news.harvard.edu/gazette/story/2017/04/over-nearly-80-years-harvard-study-has-been-showing-how-to-live-a-healthy-and-happy-life/ (accessed on).
2. Snoring—Overview and Facts. Available online: http://sleepeducation.org/essentials-in-sleep/snoring/overview-and-facts (accessed on 12 June 2024).
3. Khan, T. A deep learning model for snoring detection and vibration notification using a smart wearable gadget. *Electronics* **2019**, *8*, 987.
4. Shin, H.; Cho, J. Unconstrained snoring detection using a smartphone during ordinary sleep. *Biomed. Eng. Online* **2014**, *13*, 116.
5. Xie, J.; Aubert, X.; Long, X.; van Dijk, J.; Arsenali, B.; Fonseca, P.; Overeem, S. Audio-based snore detection using deep neural networks. *Comput. Methods Programs Biomed.* **2021**, *200*, 105917.
6. Li, R.; Li, W.; Yue, K.; Zhang, R.; Li, Y. Automatic snoring detection using a hybrid 1D–2D convolutional neural network. *Sci. Rep.* **2023**, *13*, 14009.
7. Shen, F.; Cheng, S.; Li, Z.; Yue, K.; Li, W.; Dai, L. Detection of snore from OSAHS patients based on deep learning. *J. Healthc. Eng.* **2020**. https://doi.org/10.1155/2020/8864863.
8. Mitilineos, S.A.; Tatlas, N.A.; Korompili, G.; Kokkalas, L.; Potirakis, S.M. A real-time snore detector using neural networks and selected sound features. *Eng. Proc.* **2021**, *11*, 8.
9. Ansari, M.W.; Rajak, A.; Basak, R. A Deep Learning Model to Snore Detection Using Smart Phone. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; pp. 1–5.
10. Dong, H.; Wu, H.; Yang, G.; Zhang, J.; Wan, K. A multi-branch convolutional neural network for snoring detection based on audio. *Comput. Methods Biomech. Biomed. Eng.* **2024**, 1–12. https://doi.org/10.1080/10255842.2024.2317438.
11. Yang, C.H.; Kuo, Y.M.; Chen, I.C.; Lin, F.M.; Chung, P.C. A Machine-Learning-Based Detection Method for Snoring and Coughing. *J. Internet Technol.* **2022**, *23*, 1233–1244.
12. Luo, H.; Li, H.; Lu, Y.; Lin, X.; Zhou, L.; Wang, M. Design of embedded real-time system for snoring and OSA detection based on machine learning. *Measurement* **2023**, *214*, 112802.
13. Nicla Voice. Available online: https://store-usa.arduino.cc/products/nicla-voice (accessed on 25 May 2024).
14. Snoring Dataset. Available online: https://www.kaggle.com/datasets/tareqkhanemu/snoring (accessed on 20 April 2024).
15. Audio Syntiant. Available online: https://docs.edgeimpulse.com/docs/edge-impulse-studio/processing-blocks/audio-syntiant (accessed on 30 May 2024).
16. EdgeImpulse. Available online: https://edgeimpulse.com/ (accessed on 21 April 2024).