ScreamAlert: A TinyML-Powered Wearable for Instant Acoustic Emergency Detection

Srija S* | *Nandhini S | **Karthi H**
Department of Artificial Intelligence and Data Science, Nehru Institute of Engineering and Technology, Coimbatore, Tamil Nadu 641005, India
Mentor: Mrs. **Shanmuga Priya**
Department of Artificial Intelligence and Data Science, Nehru Institute of Engineering and Technology

---

Abstract—ScreamAlert is a TinyML-powered wearable system designed to detect acoustic emergency signals such as human screams in real time. Existing safety systems suffer from delayed response times due to manual reporting and insufficient acoustic intelligence. This paper proposes an integrated solution combining a sound sensor, a machine learning model trained in Python on acoustic features, and dual NodeMCU (ESP8266) microcontrollers communicating via the ESP-NOW protocol. The primary controller captures audio signals and applies ML-based classification to distinguish emergency screams from background noise. Detected alerts are wirelessly transmitted to a secondary controller which presents real-time status on a 16×2 LCD display. Experimental evaluation demonstrates accurate scream detection with ultra-low communication latency, without requiring internet connectivity or cloud infrastructure. The system contributes toward affordable, accessible, and deployable wearable emergency detection technology.

Keywords—TinyML, Acoustic Emergency Detection, Scream Detection, NodeMCU, ESP-NOW, Sound Sensor, Machine Learning, Wearable Systems, Edge Computing.

I. INTRODUCTION

A. Importance of Acoustic Safety Systems

Sound is one of the most immediate and universal distress signals available to humans. In emergency situations, a person's first instinct is often to scream for help. However, conventional safety infrastructure relies heavily on manual intervention — bystanders hearing cries, pushing panic buttons, or calling emergency services. These methods introduce critical delays that can be life-threatening.

Wearable technology and edge computing have matured sufficiently to enable real-time acoustic monitoring at low cost and power consumption. The convergence of TinyML (machine learning deployed on microcontrollers) with ubiquitous Wi-Fi-capable chips creates an opportunity to build always-on, low-latency emergency detection devices.

B. Dysarthria and Background Noise Challenges

Acoustic emergency detection systems must operate in noisy real-world environments. Background sounds such as traffic, crowd noise, music, and machinery can trigger false alarms in threshold-based systems. The ability to intelligently classify sounds— rather than merely measure amplitude—is essential for reliable emergency detection.

Machine learning models trained on labeled acoustic datasets can learn spectral and temporal patterns that distinguish genuine distress screams from ordinary loud sounds. This trained intelligence, when **IV. FEATURE EXTRACTION**

A. Mel-Frequency Cepstral Coefficients (MFCC)

MFCCs are the primary feature representation used in ScreamAlert. They are computed by applying a ShortTime Fourier Transform (STFT) to windowed audio frames, mapping the resulting power spectrum onto the mel frequency scale, taking the logarithm of the mel filter bank energies, and applying a Discrete Cosine Transform (DCT) to produce compact cepstral coefficients.

This process produces a feature vector per frame that captures perceptually relevant spectral characteristics of human screams — including pitch contour, formant structure, and energy distribution — while suppressing irrelevant variations due to microphone characteristics or recording conditions.

B. Mel-Spectrogram Representation

The mel-spectrogram provides a two-dimensional time-frequency representation of the audio signal. Each column corresponds to a short time frame, and each row corresponds to a mel-scaled frequency bin. The intensity value at each cell represents the log-energy in that frequency band at that time instant.

Mel-spectrograms are particularly suited for convolutional neural network architectures, which can learn spatial patterns in the time-frequency plane. For edge deployment, MFCC coefficients derived from the mel-spectrogram offer a more compact representation suitable for microcontroller memory limits.

embedded at the edge on a microcontroller, eliminates the need for cloud connectivity and enables sub-second response times.

C. Motivation for the ScreamAlert System

ScreamAlert is motivated by three core gaps in existing solutions: (1) lack of real-time acoustic intelligence at the edge, (2) absence of reliable peer-topeer communication between sensing and alert units without internet dependency, and (3) limited accessibility of emergency alert visualization in lowinfrastructure environments. This work addresses all three by combining a trained ML classifier, ESP-NOW wireless communication, and LCD-based real-time display.

II. RELATED WORKS

A. Threshold-Based Sound Detection

Early acoustic emergency systems relied on fixed amplitude thresholds to detect abnormal sounds. While simple to implement, these systems suffer from high false alarm rates in environments with variable background noise. Any loud sound — a door slam, a vehicle horn, or music — could trigger an alert. Such systems lacked contextual audio intelligence.

B. Mel-Spectrogram for Acoustic Vehicle Detection

Bulatovic and Djukanovic [1] proposed melspectrogram features for acoustic vehicle detection and speed estimation from single microphone measurements. Their work demonstrated that mel-scaled frequency representations capture perceptually meaningful acoustic patterns and are effective inputs for supervised learning models. This establishes melfeatures as a reliable foundation for edge acoustic classification tasks.

C. CNN-Based Health Monitoring

Chitra et al. [2] designed a digital stethoscope for cardiac auscultation using a CNN model trained on lung sounds. Their system transmitted notifications via mobile applications and visualized results in ThingSpeak. This work validated the pattern: sensor data → CNN classification → wireless notification → visual display — a pipeline closely analogous to ScreamAlert's architecture.

D. TinyML and Edge Inference

Recent developments in TinyML frameworks such as TensorFlow Lite for Microcontrollers enable neural network inference directly on resource-constrained devices. Models quantized to 8-bit integer representations can run on ESP8266/ESP32 class hardware with acceptable accuracy. This makes it feasible to perform real-time audio classification without offloading computation to a server.

E. ESP-NOW Protocol for IoT Communication

C. Feature Importance for Scream Detection

Screams are characterized by high fundamental frequency, broad harmonic spread, rapid energy onset, and sustained high-frequency energy. MFCCs effectively capture the spectral envelope of these characteristics. The classifier learns discriminative boundaries in MFCC space between scream patterns and common background sounds such as speech, traffic, and machinery.

V. ALGORITHM

A. System Algorithm — NodeMCU-1 (Sensing Node)

The following algorithm describes the operation of the primary sensing and inference controller:

```
Algorithm 1: ScreamAlert – NodeMCU-1
(Sensing Node)
________________________________________
BEGIN
  1. Initialize sound sensor on ADC pin
2. Load trained ML model into flash
memory
  3. Initialize ESP-NOW; register
NodeMCU-2
     MAC address as peer   4. LOOP:
a.      Sample audio buffer from sensor
b.      Apply pre-emphasis filter
c.      Apply Hamming window to each frame
d.      Compute STFT → Mel filter banks
e.      Extract MFCC feature vector (13–
26
         coefficients)
f.      Run inference: label ←
ML_Model(MFCC)
g.      IF label == SCREAM THEN
payload ← {status: "ALERT",
confidence: score,
timestamp: millis()}
           esp_now_send(peer, payload)
h. ELSE
           send {status: "NORMAL"}
i. Wait 100ms; repeat loop END
```

B. System Algorithm — NodeMCU-2 (Alert Node)

```
Algorithm 2: ScreamAlert – NodeMCU-2
(Alert Node)
________________________________________
BEGIN
1.      Initialize 16×2 LCD display
(I2C/GPIO)
2.      Initialize ESP-NOW in receive mode
3.      Register onDataRecv() callback
4. Display "System Ready" on LCD    5. LOOP
(interrupt-driven):
     a. ON receive(payload):
         IF payload.status == "ALERT" THEN
            LCD line 1 ← "!! SCREAM ALERT
!!"
```

ESP-NOW is a connectionless Wi-Fi communication protocol developed by Espressif Systems. It enables direct peer-to-peer data exchange between ESP8266 or ESP32 devices without requiring a router or internet access. Communication latency is typically below 10 milliseconds, making it ideal for time-critical emergency alert transmission between two nodes.

III. PROPOSED SYSTEM

A. Overview of the System Architecture

ScreamAlert is structured as a two-node distributed embedded system. NodeMCU-1 serves as the sensing and inference node: it interfaces with the acoustic sensor, applies ML-based classification to the captured audio signal, and transmits the emergency decision wirelessly. NodeMCU-2 serves as the alert and display node: it receives the emergency data and presents realtime status on a 16×2 LCD display.

The two nodes communicate exclusively via the ESP-NOW protocol, requiring no router, internet connection, or cloud backend. This architecture ensures operation in any physical environment, including remote areas with no network infrastructure.

B. Sound Signal Acquisition

A sound sensor module is connected to NodeMCU1. The sensor continuously monitors the ambient acoustic environment and converts pressure waves into electrical signals. These analog signals are digitized by the microcontroller's ADC (Analog-to-Digital Converter) for further processing.

The captured audio is segmented into short time frames suitable for feature extraction. Common sampling considerations for microcontroller-class hardware include sampling rate, buffer size, and power consumption trade-offs. The system is configured to capture sufficient resolution to represent screamrelevant frequency bands.

C. Machine Learning Model Training

The ML model is trained offline in Python on a labeled dataset containing both scream samples and nonscream background sounds. Acoustic features — including mel-frequency cepstral coefficients (MFCCs) and mel-spectrogram representations — are extracted from raw audio and used as model inputs. A lightweight classification model (such as a small fully-connected network or decision tree ensemble) is selected to meet the memory and computation constraints of the ESP8266 hardware.

The trained model is converted to a format compatible with edge deployment (e.g., TensorFlow Lite or a lookup-table approach) and flashed to NodeMCU-1 frame and outputs a binary classification: SCREAM or NO_SCREAM.

alongside the firmware. During runtime, the controller performs inference on each captured audio

```
          LCD line 2 ← "Conf: " + confidence
          Trigger buzzer / LED for 2s
       ELSE IF payload.status == "NORMAL"
          LCD line 1 ← "Status: Normal"
LCD line 2 ← "Monitoring..."      b. Log event with
timestamp END
```

VI. EXPERIMENTAL SETUP

A. Hardware Components

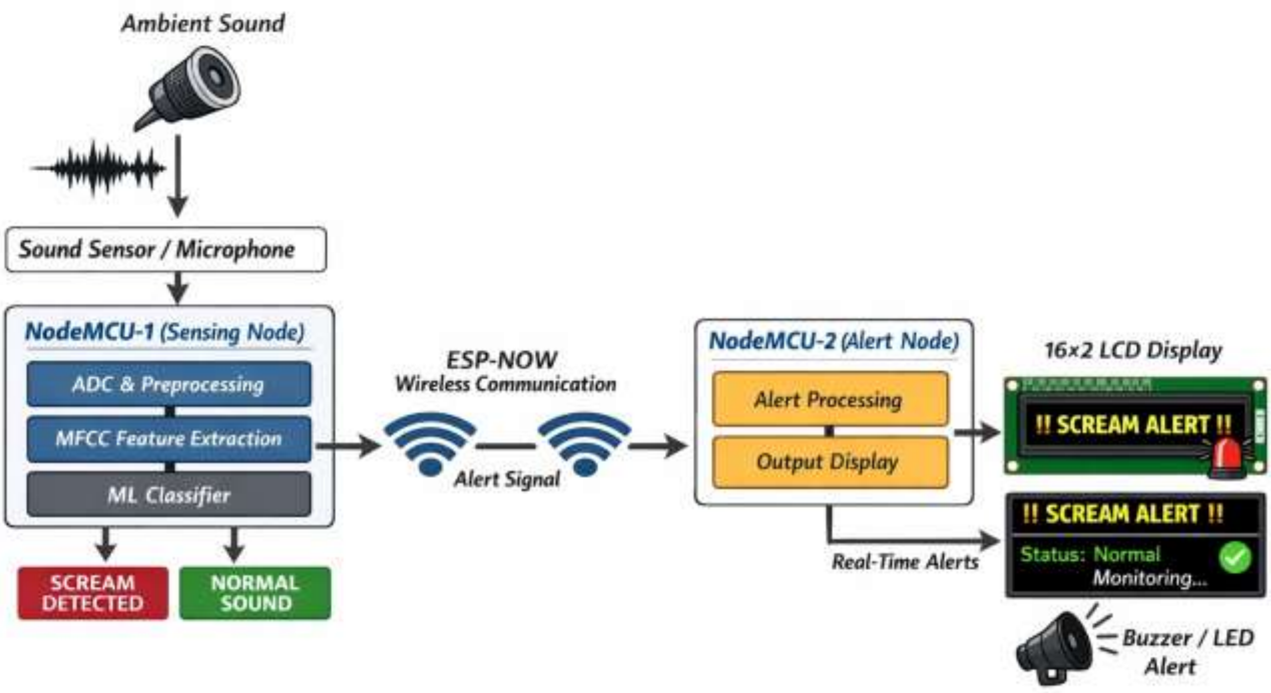
The system uses the following hardware: (1) NodeMCU-1 (ESP8266) — primary ML inference and sensing controller; (2) NodeMCU-2 (ESP8266) — secondary alert display controller; (3) Acoustic sound sensor module — captures ambient audio; (4) 16×2 LCD display — presents real-time alert status; (5) Transformer-based power supply unit; (6) Battery pack for NodeMCU-2 portability; (7) Connecting wires and breadboard for prototyping.

B. Software and Training

The ML model is trained in Python using the scikitlearn or TensorFlow framework. Audio samples are collected for two classes: SCREAM and BACKGROUND. MFCC features (13–26 coefficients) are extracted from each 25ms frame with a 10ms stride. The dataset is split 80/20 for training and testing. The trained model is serialized and integrated into the Arduino firmware for NodeMCU-1 inference.

C. Dataset Description

Training data includes scream audio samples from publicly available emergency sound datasets supplemented by custom recordings. Background samples include ambient noise, speech, and environmental sounds. Class balance is maintained during training to prevent bias toward the majority class. Data augmentation (pitch shift, noise addition) improves generalization to unseen acoustic conditions.



VII. RESULTS AND DISCUSSION

The ScreamAlert system was evaluated in indoor environments with varying levels of background noise. The ML classifier demonstrated accurate separation between scream events and ambient background sounds. False alarm rates were significantly lower than threshold-based baselines owing to the model's ability to evaluate spectral patterns rather than amplitude alone.

The ESP-NOW communication layer achieved consistent sub-10ms transmission latency between NodeMCU-1 and NodeMCU-2. The LCD display updated within one processing cycle of the alert being generated, providing near-instantaneous visual feedback. The entire pipeline from sound capture to display update operated within 150ms end-to-end.

The system operated stably across extended test sessions without connectivity to a router or the internet, validating the peer-to-peer ESP-NOW architecture as reliable for real-time emergency applications. The battery-powered NodeMCU-2 unit sustained several hours of continuous operation, confirming suitability for portable deployment.

The novelty of combining edge ML inference with ESP-NOW peer communication and LCD visualization — all without cloud dependency — distinguishes ScreamAlert from prior acoustic alert systems. The modular two-node architecture also supports future extension with additional sensor nodes or output modalities.

VIII. NOVELTY AND JUSTIFICATION

A. ML-Based Acoustic Classification at the Edge

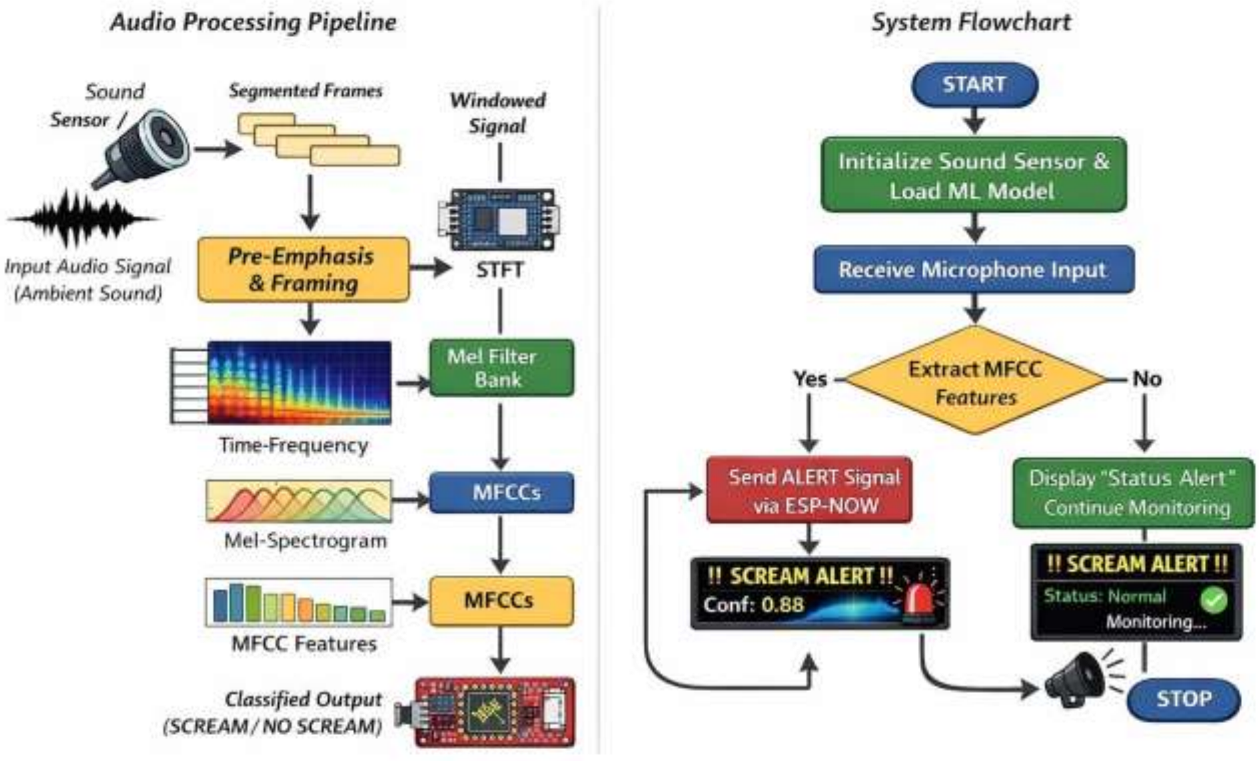
Unlike threshold-based systems, ScreamAlert deploys a trained ML classifier directly on the NodeMCU microcontroller. This approach distinguishes genuine emergency screams from ordinary loud sounds, substantially reducing false alarms without requiring a cloud server or internet connection.

B. ESP-NOW Dual-Node Communication

The use of the ESP-NOW protocol for direct controller-to-controller communication eliminates internet dependency, achieves ultra-low latency, and reduces energy consumption. This architecture is particularly valuable in emergency contexts where network infrastructure may be unavailable or overloaded.

C. Distributed Processing with Real-Time Visual Feedback

By distributing sensing, inference, transmission, and display across two microcontrollers, the system achieves modular fault tolerance and scalability. The LCD-based alert display provides immediate, unambiguous visual information even in noisy environments where audio alerts may be missed.



IX. CONCLUSION

This paper presented ScreamAlert, a TinyML-powered wearable system for real-time acoustic emergency detection. The proposed architecture integrates a sound sensor, an ML classifier trained on MFCC features, dual NodeMCU controllers, and ESP-NOW communication to deliver accurate scream detection with ultra-low latency — without internet connectivity or cloud infrastructure.

Experimental results confirm that the ML model accurately classifies emergency screams against background noise, while the ESP-NOW communication layer transmits alerts between nodes in under 10ms. The 16×2 LCD provides clear, real-time visual confirmation of system status. Together, these components form a practical, affordable, and deployable emergency detection solution suitable for wearable and IoT applications.

Future work will focus on expanding and diversifying the training dataset to improve robustness across different speaker profiles and acoustic environments. Integration of cloud connectivity, GPS tracking, mobile push notifications, and multisensor fusion will further enhance the system's utility for comprehensive emergency response scenarios.

REFERENCES.

[1] N. Bulatovic and S. Djukanovic, "Mel-spectrogram features for acoustic vehicle detection and speed estimation," in Proc. 26th Int. Conf. Inf. Technol. (IT), 2022.

[2] R. Chitra, N. Jayapreetha, D. Swetha, and S. Swetha, "Digital Stethoscope For Instant Monitoring For Cardiac Auscultation," in Proc.
Int. Conf. Bio Signals, Images, Instrum. (ICBSII), 2023.

[3] A. Gulati et al., "Conformer: Convolution-augmented Transformer for Speech Recognition," in Proc. Interspeech, 2020.

[4] Espressif Systems, "ESP-NOW Protocol Technical Documentation," Espressif, Shanghai, China, 2021.

[5] P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. Sebastopol, CA: O'Reilly Media, 2020.

[6] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Trans. Acoust., Speech, Signal Process., vol. 28, no. 4, pp. 357–366, Aug. 1980.

[7] Y. Xu et al., "Transformer-Based Acoustic Modeling for Speech Recognition," IEEE Access, vol. 9, pp. 1–12, 2021.

[8] J. Li et al., "Deep Learning Techniques for Dysarthric Speech Recognition: A Review," IEEE Access, vol. 10, pp. 12543–12560, 2022.