Embedded IoT System for Acoustic Precipitation Phase Partitioning via Edge ML and MFCCs

Soheyl Faghir Hagh, *Graduate Student Member, IEEE*, Jordan Bourdeau, Julia Sober, Rachael Chertok, Christopher Jepsen, Parmida Amngostar, Lucas Levine, Casey Forey, Tian Xia, *Senior Member, IEEE*, and Christian Skalka

***Abstract*—Accurate and real-time detection of precipitation phases is essential for hydrological modeling, water resource management, and climate impact assessments. However, conventional methods struggle to distinguish precipitation phase near freezing and are unsuitable for distributed deployment in remote, complex terrain due to cost, size, and power constraints. In this work, we present an integrated acoustic sensing system that combines machine learning (ML) on the edge with acoustic sensing and Mel-frequency cepstral coefficients (MFCCs)-based feature extraction, all implemented on our designed edge device. The device is low-cost, with LAN and WAN capabilities for near real-time reporting and can integrate multiple sensors for snow hydrology studies, in addition to detecting the precipitation phase. We present the design, fabrication, and validation of the Advanced Unified Rainfall and Atmospheric Monitoring (AURA) system, integrating sensors for temperature, humidity, ultrasonic snow depth, solar radiation, and wind velocity. AURA features an embedded ML algorithm that enables accurate classification of precipitation phases. Short-time Fourier Transform (STFT) and MFCCs are computed on the edge from recorded precipitation acoustics via novel Mel filter bank computation methods. Using support vector machine (SVM) and random forest (RF) classifiers, the RF model achieves testing accuracy of 97.35% on simulated precipitation acoustics and 85.67% in consolidated class (CC) environmental recordings. The SVM classifier achieves 98.07% accuracy on simulated acoustics and 85.99% on CC environmental recordings. The low-power AURA network uses a long range (LoRa) star topology with time-division multiple access (TDMA) and an Iridium gateway for reliable data transfer from remote sites. Signal-to-noise ratio (SNR) analysis and comprehensive field tests confirm and validate system performance.**

Received 26 August 2025; accepted 17 September 2025. Date of publication 22 September 2025; date of current version 8 December 2025. This work was supported by the Cooperative Institute for Research to Operations in Hydrology (CIROH) under Award AWD00001596. *(Corresponding author: Christian Skalka.)*

Soheyl Faghir Hagh and Parmida Amngostar are with the Department of Electrical and Biomedical Engineering, University of Vermont, Burlington, VT 05405 USA (e-mail: sfaghirh@uvm.edu; pamngost@uvm.edu).

Jordan Bourdeau, Julia Sober, Rachael Chertok, Christopher Jepsen, Lucas Levine, and Casey Forey are with the Department of Computer Science, University of Vermont, Burlington, VT 05405 USA (e-mail: jbourde2@uvm.edu; jsober@uvm.edu; rchertok@uvm.edu; cdjepsen@uvm.edu; ljlevine@uvm.edu; cforey@uvm.edu).

Tian Xia is with the Department of Electrical and Biomedical Engineering and the Water Resources Institute, University of Vermont, Burlington, VT 05405 USA (e-mail: txia@uvm.edu).

Christian Skalka is with the Department of Computer Science and the Water Resources Institute, University of Vermont, Burlington, VT 05405 USA (e-mail: ceskalka@uvm.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2025.3613138, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2025.3613138

***Index Terms*—Acoustic sensing, long range (LoRa), machine learning (ML) on the edge, Mel-frequency cepstral coefficients (MFCCs), precipitation phase partitioning, signal-to-noise ratio (SNR), time-division multiple access (TDMA).**

I. INTRODUCTION

PRECIPITATION phase partitioning is the classification of various types of precipitation, such as rain, snow, sleet, and hail, and the quantification of their respective contributions to the total amount of precipitation [1]. Identifying the precipitation phase accurately is crucial in hydrology, climate science, and water resource management, as it determines how water is stored and released in a given environment [2]. With climate change and its impacts on precipitation patterns [3] and transformations into terrestrial water storage, accurately monitoring precipitation phase transitions is increasingly important to assess flood and drought risks and predict long-term impacts on regional and global water resources [4].

Current precipitation monitoring techniques rely on meteorological measurements and threshold methods [5], sophisticated remote sensing and instruments such as laser and radar disdrometers [6], microwave attenuation measurement, and commercial wireless measurement systems [7]. Laser and radar disdrometer networks are utilized to enhance spatial resolution [8], [9]. Meteorological observation satellites offer significant value, but their spatial and temporal resolutions often do not meet the specific needs for high-resolution monitoring [10]. Cui et al. [11] demonstrated that basin-scale wireless sensor networks can estimate rain–snow transition elevations. In mountainous or remote regions, where traditional methods suffer from coverage gaps and reduced accuracy, there is a need for localized, real-time monitoring systems.

Recently, acoustic detection in environmental monitoring has gained significant attention due to its potential in various applications such as acoustics, signal-based, and machine learning (ML)-assisted rainfall and intensity detection [12], [13], [14], [15], [16], [17], [18], as well as snow sensing systems [19]. A widely used technique for analyzing acoustic signals is the extraction of Mel-frequency cepstral coefficients (MFCCs) as features.

MFCCs are features widely used in audio signal processing that represent the short-term power spectrum of a sound, based on a perceptual scale that mimics the human ear [20].

Originally developed for speech recognition, MFCCs have since been applied in a variety of acoustic classification tasks, including environmental sound classification such as forest soundscapes, animal vocalizations, acoustic snow sensing, and wildlife monitoring [21], [22], [23], [24], [25]. MFCCs capture the overall shape of the signal spectrum. Based on the Mel scale, they mimic human hearing and are computationally efficient, compressing spectral data into fewer features. The MFCC application is further extended to precipitation phase detection [26]. These advances reflect a growing interest in leveraging acoustic signals as a non-invasive and cost-effective approach to monitor and interpret environmental conditions. Further advancement is required in developing reliable, low-cost, real-time precipitation phase monitoring sensor systems capable of operating in remote and harsh environments. MFCCs are often used because they outperform other acoustic features [27].

The significant growth of Internet of Things (IoT) devices has enabled and intensified demand for ML and computing at the edge [28], [29], [30], [31]. Instead of offloading sensor data to the cloud (which adds latency and bandwidth costs), emerging approaches run ML inference locally on devices or nearby gateways. For instance, recent work in acoustic sensor networks shows that tiny wireless nodes can locally recognize sound events using onboard convolutional neural network (CNN) models [32]. Similarly, in environmental sensing, precipitation detection has been enhanced by edge intelligence. IoT sensors can classify rainfall versus snowfall in real time using lightweight ML [33], [34], [35], [36], avoiding the need to stream raw data. In general, performing edge analytics yields markedly lower latency, improved bandwidth efficiency, and enhanced data privacy compared to only cloud-based methods. Moreover, modern TinyML techniques have made it possible to run optimized neural network models with limited CPU and memory resources, bringing intelligent analytics to even the most resource-constrained IoT nodes [37].

To enhance precipitation sensing efficiency through advances in IoT technology, we propose a novel multimodal wireless acoustic system that leverages a support vector machine (SVM) classifier and MFCC feature extraction on the edge device to provide real-time multitype precipitation phase prediction. The proposed system incorporates additional meteorological sensors, especially relevant to remote snow hydrology, including a temperature and relative humidity sensor, an ultrasonic-based snow depth sensor, and an illuminance (light) sensor with an anemometer for wind velocity measurement. The system features low-power wide-area network (LPWAN) technology using long range (LoRa) and satellite communications for near-real-time (NRT) transmissions. In this work, we design and fabricate an integrated advanced unified rainfall and atmospheric monitoring (AURA) circuit board that features an ATmega 2560 microcontroller unit (MCU) and its peripherals, including sensors, input/output (I/O) pins, power distribution switches (PDSs) for sensors, and the RockBlock Iridium 9603 modem.

In the star LoRa network, one AURA board is configured as the gateway node, which communicates with the Iridium satellite network for NRT data transmission. Additionally, we incorporate a time-division multiple access (TDMA) scheme to control the wireless (LoRa) data transmission between the gateway and all leaf sensor nodes (peripheral nodes). A micro-SD card circuit is designed to provide onboard data storage, and a real-time clock (RTC) provides accurate timestamps for each node. The system operates with low power consumption, integrating edge computing for real-time processing, eliminating dependency on cloud-based computation.

Our work is novel in both algorithmic efficiency and system design. To address the intense memory and computational needs for MFCC extraction, we introduce two lightweight methods for applying Mel filters: a lookup table and a compact multiplication-based linear model, both of which are stored in the program memory of microcontrollers, such as the AVR family. Furthermore, we present a low-cost, low-power edge device that performs real-time acoustic precipitation classification, entirely on board, using MFCC features and ML. With both LAN and LoRaWAN connectivity, the system supports scalable deployment across remote sensor networks, making it well-suited for snow hydrology and environmental monitoring in complex terrains.

To provide proof-of-concept that ML models can accurately detect precipitation phase and intensity, we have developed both simulated precipitation acoustics datasets and collected real (naturally occurring) precipitation acoustic data during Winter 2025. We used these to train a variety of lightweight models that are viable for implementation on the AURA platform. Among these, SVM and random forest (RF) models with MFCC inputs had the best performance. Given its smaller footprint, our best SVM model was chosen for the prototype system described in this article, but the use of RF or even small neural network models is feasible. Our best SVM model had an overall testing accuracy of 98.07% and 85.99% for the simulated and actual environmental acoustics, respectively. Overall, the complete system we describe here demonstrates the feasibility of low-power embedded devices for real-time environmental monitoring. The combination of acoustic sensing with traditional meteorological data offers a novel, effective solution for precipitation phase classification, paving the way for improved hydrological modeling and climate impact assessments.

The remainder of this article is structured as follows. Section II presents the system architecture and hardware design. Section III details the digital signal processing and time- and frequency-domain feature extraction methodology. Section IV details the SNR analysis and MFCC-DFT robustness. Section V describes the edge-embedded ML and classifications. Section VI provides details on IoT sensor networks, followed by calibration results in Section VII. Field deployment considerations are elaborated in Section VIII, followed by results and discussion in Section IX, and prediction mismatch, comparison, and conclusion in Section X.

II. SYSTEM ARCHITECTURE

In this section, we detail the device architecture and design of the main controller circuit, which runs ML and signal processing and controls the sensors. The circuit features LoRa

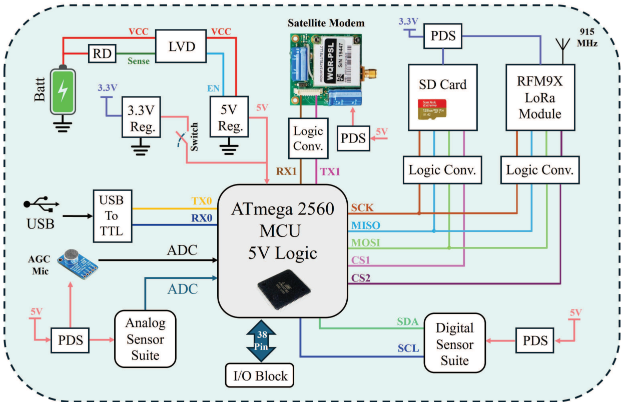

Fig. 1. Block diagram illustration of the sensor system architecture (the satellite modem is only for the gateway node). The figure highlights LVD, resistive divider (RD), PDSs, bidirectional 5–3.3-V logic converters, voltage regulators, USB to TTL converter, Atmega 2560 MCU, and sensors and peripherals.

and LPWAN protocols, which include bidirectional node-to-node and node-to-gateway communication.

A. *Embedded Sensor Circuit Design*

In this section, we describe the design sections of the proposed integrated edge sensor system. As depicted in Fig. 1, the designed edge device is built around an Atmega 2560 MCU, which controls multiple peripherals, including the sensors, PDS, satellite modem, micro-SD card, and LoRa module.

1) Power Supply and LVD: We use two low-dropout (LDO) linear voltage regulators (LVRs) to supply 5-V and 3.3-V for circuit operation, as shown in Fig. S1. The 5-V regulator powers the ATmega 2560 MCU, peripherals, and the 3.3-V regulator. A low-voltage disconnect (LVD) circuit, designed with a TLV6713 micropower comparator with an internal reference voltage of 400 mV and hysteresis, drives the active-high enable pin of the 5-V regulator. The LVD disables the entire circuit as soon as the input battery voltage falls below a configurable threshold. The input–output voltage difference on an LVR should be minimized while considering the dropout voltage, as in the following equation, to reduce power dissipation in the regulators:

$$P_{\text{loss}} = (V_{\text{in}} - V_{\text{out}}) \cdot I_{\text{out}}. \tag{1}$$

2) Microcontroller Unit: The 5-V ATmega2560 MCU clocked at 16 MHz runs the trained ML algorithm, signal processing (audio sampling with 10-bit internal analog-to-digital converter (ADC), short-time Fourier transform (STFT) calculations, and MFCC feature extraction), LoRa communication, and scheduling tasks. The MCU depicted in Fig. S2 features 256 KB of flash program memory, 8 KB of static random access memory (SRAM), and 4 KB of electrically erasable programmable read-only memory (EEPROM), providing sufficient storage for program code and signal processing.

3) RTC and SD Card: The AURA board adopts a DS3231 RTC chip and an SD card socket. The RTC provides accurate timestamps for measurements and synchronization with an on-chip temperature sensor. The SD card (Fig. S3) stores the logged data locally on each node as backup. Since SD cards operate at 3.3 V, a level shifter is required for serial peripheral interface (SPI) communication of the SD module with the 5-V MCU.

TABLE I

AURA SYSTEM POWER CONSUMPTION

State (mode)	Peak Current (mA)	Peak Power (W)[a]
DSP	76.4	0.916
Audio Rec.[b]	94.2	1.130
MB7060 Sonar[c]	59	0.708
Solar Rad.[c]	56	0.672
Anemometer[c]	66	0.792
DHT22[c]	62	0.744
LoRa (TX, RX)[d]	128.21 ea.	1.538 ea.
9603 Sat. (TX)[d]	150.2	1.802

[a] Measured peak DC power draw from a 12 V source; actual use is duty-cycled (i.e., the device is turned on periodically).
[b] Includes mic activation, recording, and SD write.
[c] Sensor current includes switch parasitic current and analog read.
[d] TX and RX stand for transmit and receive, respectively.

4) Universal Serial Bus: To program the MCU, a universal serial bus (USB)-to-serial transistor-to-transistor logic (TTL) programmer, illustrated in Fig. S4, featuring a Type-C USB connector, is utilized. The circuitry incorporates an automatic voltage selector circuit using a Schottky diode ($V_f = 0.39\,V$) that provides a voltage source for operation when powered via USB. A polymeric positive temperature coefficient (PTC) thermistor (up to 500 mA) is integrated as a resettable fuse to protect the computer connected to the board upon short circuits.

5) Sensors and PDSs: The sensor suite includes six meteorological sensors and four additional connectors for sensor expansion, that is, extra microphones and a barometric pressure sensor. The default sensors include an Adafruit MAX 9814 microphone [set to 40-dB gain with automatic gain control (AGC)], an FDS100 photodiode used for the illuminance sensor, and an MB7092 ultrasonic range-finder (10-Hz refresh rate) to measure snow depth in addition to a DHT22 sensor to read temperature and relative humidity. We utilize an Adafruit anemometer sensor to quantify wind velocity. The circuit facilitates four additional sensor connectors for sensor expansion. Power to the peripherals is managed by two TPS2087 devices, quad-channel high-side PDSs with active-low enable inputs, as shown in Fig. S1.

6) Power Consumption Profiling: Table I summarizes the peak active current of AURA board components. The board is powered by a 12-V sealed lead-acid battery. In deep sleep operation mode, only the MCU draws 260-μA current. In the LVD mode, the current drops to 140 μA, with only the LVD comparator being active [Fig. S14].

We estimate that under ideal conditions with a 5% monthly self-discharge rate for a typical sealed lead-acid battery, with an average current draw of 20.2 mA, the device should run for approximately 15.2 days without solar panels and recharging. We add a 10-w solar panel to replenish the battery energy during the day [refer to battery life estimation in the supplementary material].

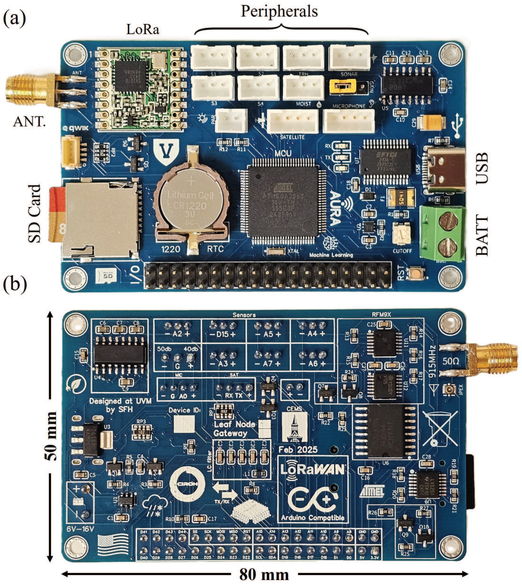

Fig. 2. AURA control board (quad-layer) is an Arduino-compatible edge device supporting the LoRaWAN protocol (designed at UVM). (a) Top view: includes sensor, SD card, satellite, battery, USB port, and I/O pins. (b) Bottom side view. Fabricated AURA PCB price: about 110 (USD).

7) Bidirectional Level Shifters: Peripherals operating at different logic levels (i.e., 3.3 V), such as the SD card, Quick Wire Inter-Integrated Circuit (QWIIC), RFM9X LoRa transceiver, and RockBlock 9603 modem, require bidirectional logic converters (3.3–5 V) to communicate with the MCU. We use an N-channel metal-oxide semiconductor field-effect transistor (MOSFET) level shifter for QWIIC and RockBlock 9603 communication. For the SD card and LoRa modem, which rely on SPI communication, we implement quad-channel level shifters to shift four signal levels (CS, MOSI, MISO, and SCK).

8) LoRa Radio Transceiver: The AURA circuit board incorporates a 915-MHz RFM9X LoRa transceiver, which produces up to +20 dBm transmit power and supports transmission distances of up to 1.23 km in an urban area. The transceiver handles data packets, error correction, and retransmission, enabling LoRa, low-power communication via SPI. The RFM9X schematic is provided in Fig. S3.

B. Embedded Printed Circuit Board Design and Fabrication

Fabricated utilizing FR-4 material with a quad-layer board (1.6-mm thick), the top and bottom views of the AURA board PCB are shown in Fig. 2(a) and (b). The PCB layout is provided in Fig. S5.

III. DIGITAL SIGNAL PROCESSING

This section details the audio sampling and processing of the STFT and MFCCs on the edge device (ATmega 2560 microcontroller).

A. Audio Sampling and Discretization

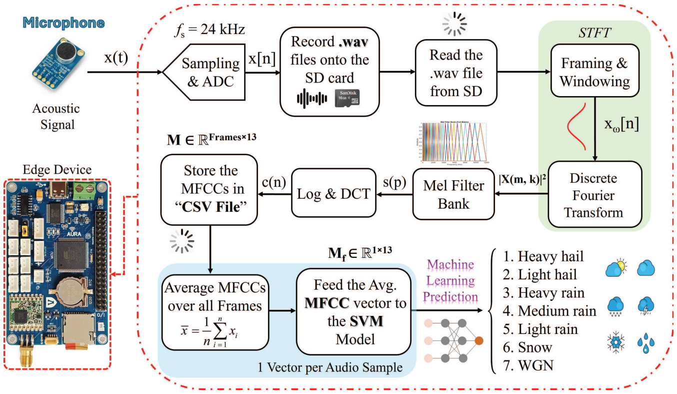

Fig. 3. Full signal processing and feature extraction pipeline on the edge device (ML on the edge).

The microprocessor samples the audio signal from the microphone using AGC with an ADC at a sampling rate of 24 kHz (8-bit ADC mode), accurately capturing frequencies up to half of the bandwidth (12 kHz), as per the Nyquist theorem. The audio is recorded in 8-bit format using the TMRpcm library for a time length of 10 s to ensure that sufficient precipitation events are captured. Each analog audio sample is then quantized to discrete digital values and sequentially stored in an uncompressed Pulse Code Modulation (PCM) format. The sampled data are then written to the SD card in the Waveform Audio (WAV) File Format, which consists of a 44-byte header followed by raw audio samples. The digital signal-processing flow is depicted in Fig. 3.

B. STFT and MFCC Extraction

MFCCs provide a compact representation of spectral envelopes of audio signals by modeling the characteristics of human auditory perception [38]. They are derived from a time–frequency representation computed via the STFT.

Let $x[n]$ be a discrete-time audio signal of length L. To extract overlapping frames, we apply a window $w[n]$ of length N to segments of $x[n]$ with a hop size H, where $L, N \in 2\mathbb{Z}^+$ (positive even integers), specifically chosen as 2^q, $q > 0$ (positive powers of two) for computation using the Cooley–Tukey radix-2 DFT. Zero-padding is applied to the final samples of $x[n]$ if $L \notin 2^q$ to ensure L conforms to the required length. The total number of frames can be calculated by the following equation:

$$M = \left\lfloor \frac{L - N}{H} \right\rfloor + 1. \tag{2}$$

To reduce the spectral leakage, each frame is then windowed using the Hamming window function defined in the following equation:

$$w[n] = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N - 1}\right), \quad 0 \le n \le N - 1. \tag{3}$$

The discrete STFT is computed using the following equation:

$$X(m, k) = \sum_{n=0}^{N-1} x[n + mH]\, w[n]\, e^{-j\frac{2\pi k n}{N}} \tag{4}$$

where m indexes the time frames and k indexes the frequency bins (typically, k ranges from 0 to $\frac{N}{2}$ assuming $N \in 2^q$ corresponding to the Nyquist frequency bins). The magnitude of $X(m, k)$ is typically visualized as the signal spectrogram, which shows how the frequency content of $x[n]$ evolves. The STFTs extracted via MATLAB's built-in functions and our approach are illustrated in Fig. S6.

In the MFCC extraction pipeline, $|X(m, k)|^2$ is utilized to compute the spectral power, which is then filtered using the Mel filter bank. The pre-emphasis step is omitted for precipitation acoustics because such signals do not exhibit the strong high-frequency attenuation seen in speech.

C. Mel Filterbank

Mel filter banks are designed using the nonlinear Mel scale to reflect human auditory perception. Mel filters are applied to the DFT power spectrum $|X(m, k)|^2$ within a linear frequency domain. The conversion from linear frequency f (in Hz) to the Mel scale is given by the following equation:

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right). \tag{5}$$

The Mel scale $M(f)$ is applied to evenly space filter centers in the Mel domain. The frequencies are first mapped to Mel space, evenly spaced in Mel space, then converted back to Hz and multiplied by DFT bins. The normalized triangular Mel filter bank $H_p(k)$ is defined over the DFT bin indices k. For each filter (indexed by p with $0 \leq p \leq P - 1$, where P is the number of filters), three bin locations f_p, f_{p+1}, and f_{p+2} are specified, forming a triangular division-based implementation of the as in the following equation:

$$H_p(k) = \begin{cases} 0, & k < f_p \text{or } k \geq f_{p+2} \\ \dfrac{k - f_p}{f_{p+1} - f_p}, & f_p \leq k < f_{p+1} \\ \dfrac{f_{p+2} - k}{f_{p+2} - f_{p+1}}, & f_{p+1} \leq k < f_{p+2}. \end{cases} \tag{6}$$

Each filter is centered at bin f_{p+1} (where the amplitude is 1), and the bins f_p correspond to actual frequencies spanning from f_L to f_H (Hz).

D. Logarithm and Discrete Cosine Transform

For each time frame m, the Mel-filtered spectral power is computed by summing the squared STFT magnitudes weighted by the corresponding Mel filter (we discard conjugate pairs) in the following equation:

$$s(p) = \sum_{k=0}^{N/2} |X(m, k)|^2 \, H_p(k), \quad 0 \leq p \leq P - 1 \tag{7}$$

where $s(p)$ represents the energy in the pth Mel filter for the given time frame. Note that this process is repeated for all time frames m on the MCU.

Then, the logarithm of these energies is taken, and the discrete cosine transform (DCT) in (8), which decorrelates energy in different Mel bands (filter bank energies), is applied to yield the MFCCs

$$\begin{cases} c(n) = \displaystyle\sum_{p=0}^{P-1} \log_{10}(s(p)) \cdot \cos\left(\frac{\pi n}{P}\left(p - \frac{1}{2}\right)\right) \\ n = 0, 1, \ldots, C - 1 \end{cases} \tag{8}$$

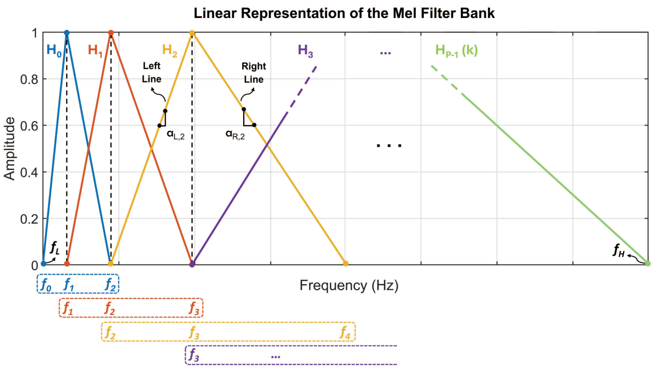

Fig. 4. Linear representation of the normalized Mel filter bank with only three filters for illustration. Overall, 30 filters are utilized in the digital signal processing.

where $c(n)$ represents the nth MFCC coefficient for the frame and C coefficients are retained. In this work, we compute the first 13 MFCCs on the edge device (ATmega 2560 MCU) from recorded precipitation acoustics.

E. Efficient MFCC Computation on Microcontrollers

While libraries such as `ArduinoFFT` and `ApproxFFT` support DFT computation on microcontrollers, MFCC extraction is constrained by RAM limitations. Applying P Mel filters to an N-point DFT requires a matrix of size $\mathbb{R}^{P\times((N/2)+1)}$, often exceeding available memory. To address this, we implement two novel memory-efficient methods: 1) a lookup table storing precomputed filter weights in program memory, and 2) a multiplication-based (computationally quicker on MCU compared to division) linear model where each triangular filter is defined by slope-intercept parameters.

1) Look-up Table: The Mel filter bank is precomputed in MATLAB and stored in program memory as a matrix $\mathbf{H}_p$ (9). This table is sparse and less efficient to store due to a high number of zero entries:

$$\mathbf{H}_p = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,\frac{N}{2}} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,\frac{N}{2}} \\ \vdots & \vdots & \ddots & \vdots \\ h_{P-1,0} & h_{P-1,1} & \cdots & h_{P-1,\frac{N}{2}} \end{bmatrix}. \tag{9}$$

2) Linear Representation: Each triangular Mel filter is defined by three frequency bins $\{f_p, f_{p+1}, f_{p+2}\}$ and modeled using two line segments in the following equation:

$$H_p(k) = \begin{cases} a_{L,p}\,k + b_{L,p}, & f_p \leq k < f_{p+1} \\ a_{R,p}\,k + b_{R,p}, & f_{p+1} \leq k < f_{p+2} \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The linear modeling of the Mel filter bank is detailed in Fig. 4. This figure outlines three full filters (for demonstration only), and the process repeats for the remaining filters. This approach has higher resolution compared to the look-up model. A total of 30 Mel filters are utilized in practical programming.

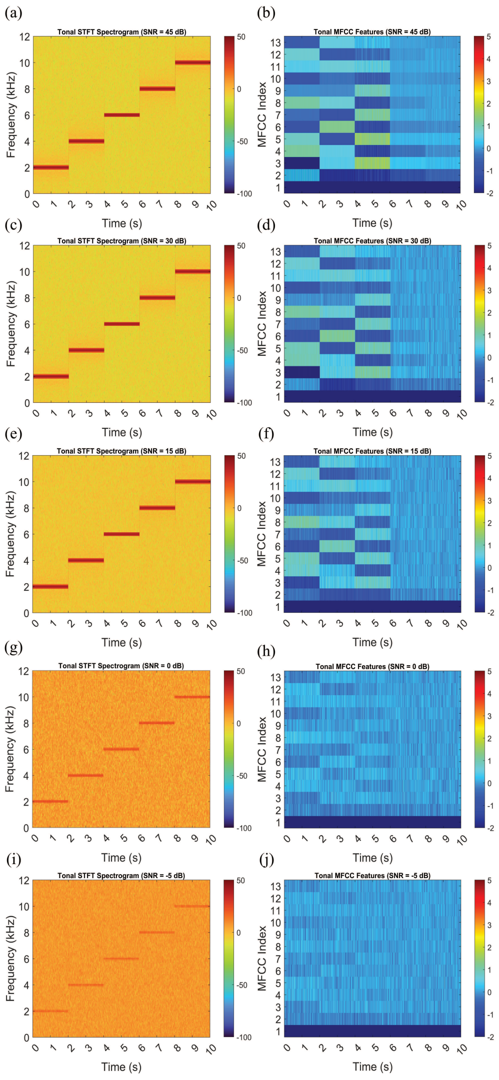

Fig. 5. STFT and MFCC spectrograms of MATLAB-generated acoustics across five SNR levels, including (a) and (b) SNR = +45 dB. (c) and (d) SNR = +30 dB. (e) and (f) SNR = +15 dB. (g) and (h) SNR = 0 dB. (i) and (j) SNR = −5 dB. Left column: STFT results; Right column: MFCC results.

Slopes and intercepts in the linear model are defined as

$$\begin{cases} a_{L,p} = \dfrac{1}{f_{p+1} - f_p}, & b_{L,p} = -a_{L,p}\, f_p \\ a_{R,p} = -\dfrac{1}{f_{p+2} - f_{p+1}}, & b_{R,p} = 1 - a_{R,p}\, f_{p+1}. \end{cases} \tag{11}$$

The Mel filter bank parameters in linear representation include $\{f_p, f_{p+1}, f_{p+2}, a_{L,p}, b_{L,p}, a_{R,p}, b_{R,p}\}$ and are stored in the program memory of the MCU. At runtime, the `pgm_read_float` function loads these values, allowing $H_p(k)$ to be computed on the fly without storing the full filter table in RAM space. In this work, we apply 30 Mel filters and a 512-point DFT (discarding the redundant conjugate points), where the lookup table approach requires $\mathbb{R}^{30\times 257}$ and the linear approach fits in $\mathbb{R}^{30\times 7}$ [Algorithm 1 in the supplementary material].

IV. SNR ANALYSIS FOR MFCC-DFT NOISE ROBUSTNESS

Signal-to-noise ratio (SNR) testing is essential for assessing the robustness of spectral features. MFCCs compress spectral energy via Mel filter banks and DFT–DCT operations. Additive noise raises the noise floor, and by injecting controlled noise at known SNRs, we evaluate how reliably DFT preserves frequency components, how stable MFCCs remain under degradation, and at what thresholds feature reliability fails.

A. Signal Generation

We synthesize a five-tone sequence with frequencies $f_i \in \{2, 4, 6, 8, 10\}$ kHz, sampled at $f_s = 24$ kHz for a total duration of $T = 10$ s. Each block has length $L = f_s D$, where $D = T/5$. The discrete-time signal $x[n]$ is constructed using (12), where phase continuity across tone transitions is preserved by the recurrence in (13)

$$\begin{cases} x[n] = A_p \sin\left(\phi_i + 2\pi f_i \dfrac{n-(i-1)L}{f_s}\right) \\ n \in [(i-1)L, iL-1], \quad i = 1, \ldots, 5 \end{cases} \tag{12}$$

where $x[n]$ is the discrete-time signal, A_p is the peak amplitude, f_i is the frequency of the ith block, L is the number of samples per block, f_s is the sampling frequency, and ϕ_i is the carry-over phase from the previous block. The block index i spans 1 to 5 (each representing a 2-s segment)

$$\phi_{i+1} = \phi_i + 2\pi f_i D, \quad \phi_1 = \frac{\pi}{2}. \tag{13}$$

Equation (13) recursively computes the starting phase ϕ_{i+1} of each block to preserve continuity.

B. Targeted SNR and Noise Injection

Let $x[n]$ be the clean signal and $y[n] = x[n] + n[n]$ the noisy signal, where $n[n]$ is white gaussian noise (WGN). The signal power P_s, noise power P_n, and resulting SNR in decibels are computed using the following equation:

$$\begin{cases} P_s = \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} x[n]^2 \\ P_n = \dfrac{1}{N}\displaystyle\sum_{n=0}^{N-1} (y[n] - x[n])^2 \\ \mathrm{SNR_{dB}} = 10\log_{10}\left(\dfrac{P_s}{P_n}\right) \end{cases} \tag{14}$$

where P_s is the average signal power over N samples, P_n is the average power of the noise, and $\mathrm{SNR_{dB}}$ is the resulting SNR in decibels.

To produce a specific SNR, the noise variance (noise power) σ_n^2 is set based on the target level using the following equation:

$$\sigma_n^2 = \frac{P_s}{10^{\mathrm{SNR_{dB}}/10}}, \quad n[n] \sim \mathcal{N}\left(0, \sigma_n^2\right). \tag{15}$$

Equation (15) determines the variance of the Gaussian noise $n[n]$ such that the resulting SNR equals the target. The noise is drawn from a zero-mean normal distribution with variance σ_n^2. Post synthesis, the true SNR is verified by re-evaluating (14) on the final WAV signal via MATLAB.

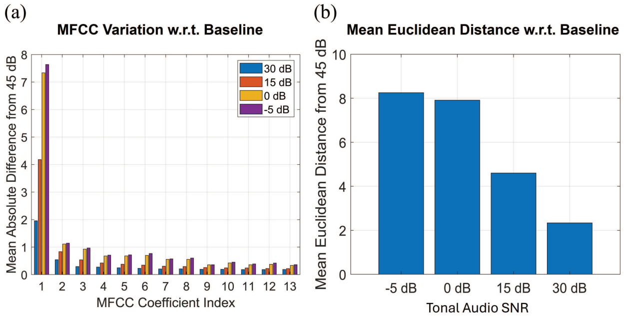

Fig. 6. MFCC variation analysis against additive WGN. (a) 2-D bar chart illustrating mean absolute MFCC index difference from baseline audio (SNR = 45 dB) across all MFCC indices. (b) 2-D bar chart showing mean Euclidean distance with respect to baseline (SNR = 45 dB).

C. Quantization Limit of WAV Storage

Exporting to b-bit PCM imposes a theoretical SNR limit due to quantization noise. The upper bound is estimated using (16), where A_{pk} is the normalized peak amplitude

$$\mathrm{SNR}_{\mathrm{quantized,dB}} \approx 6.02\,b + 1.76 + 20\log_{10}\left(A_p\right). \tag{16}$$

With $b = 8$ (8-bit audio for the TMRpcm library) and $A_p = 1$, this yields a maximum SNR of 49.92 dB. To remain well within this bound and ensure realistic comparability, we select $A_p \approx 0.57$ to result in a 45-dB SNR as the baseline reference (clean audio with no noise). We then inject different noise levels into the baseline to degrade the SNR.

D. SNR Analysis Results

Fig. 5 illustrates the five-tonal structure in both STFT and MFCC domains under increasing noise (degrading SNR). The clean audio features an SNR of 45 dB. As the SNR decreases from SNR = 30 dB to SNR = −5 dB, feature blurring and distortion increase (background noise becomes stronger), though the five-tone signature remains visible.

To quantify MFCC degradation, we apply two complementary metrics for each coefficient k: the absolute difference and the Euclidean distance. Let $M_k[n]$ and $R_k[n]$ represent the MFCC coefficient k at time frame n for the noisy and reference (clean) signals, respectively.

The average absolute difference measures individual MFCC index deviation from the reference signal (SNR = 45 dB) for each index and is defined by the following equation:

$$\Delta_k = \frac{1}{N}\sum_{n=1}^{N}\left|M_k[n] - R_k[n]\right|. \tag{17}$$

The Euclidean distance aggregates the squared deviation for each MFCC and is given by the following equation:

$$D_k = \sqrt{\sum\nolimits_{n=1}^{N}\left(M_k[n] - R_k[n]\right)^2}. \tag{18}$$

To study the effects of WGN on MFCCs, we provide two metrics, including mean absolute distance from baseline in Fig. 6(a) and mean Euclidean distance from baseline in Fig. 6(b). In Fig. 6(a), MFCC 1 shows the greatest deviation across noise levels. This coefficient represents low-order spectral shapes (e.g., energy slope and tilt) that are more susceptible to additive noise. In contrast, higher-order coefficients remain relatively stable and resilient. Fig. 6(b) presents a descending mean Euclidean distance from baseline as SNR value increases, suggesting MFCCs show less variation at lower noise levels.

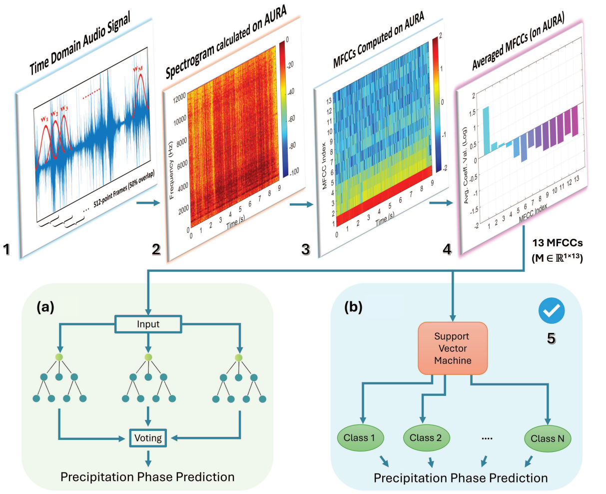

Fig. 7. MFCCs calculation steps and ML on the edge device; (a) RF and (b) SVM on the edge device (AURA board). In step 4, each color bar illustrates one averaged MFCC coefficient. All five steps are carried out on the AURA board.

E. Methodology Comparison and Validation

To compare our customized MFCC extraction functions with the standard MATLAB functions, we compute the correlation coefficient under identical conditions using a single audio file in both MATLAB and AURA. A positive correlation is found and illustrated in Fig. S7 and Table ST1 in the supplementary document.

V. EMBEDDED ML ON THE EDGE

This section details the ML (RF and SVM) classifiers trained to run on AURA using MFCC input features.

A. Acoustic Features for ML

The MFCCs extraction from a Hail audio sample and ML pipeline are illustrated in Fig. 7. The training and testing data are processed as shown in Fig.7. The 10-s recorded audio in each class is framed with a 512-sample frame width and windowed to produce the STFTs. The first 13 MFCC indices are then calculated for each audio frame and averaged over time to be utilized for training and testing of the RF model.

B. Data Collection and Labeling

An Enviro DIY Mayfly board is utilized to record 8-bit audio from a MAX9814 microphone (40 dB AGC) in a conical chamber made from aluminum. We utilize a conical geometry that makes an effective resonance chamber by naturally shedding snow and frozen precipitation due to its sloped surfaces, while also supporting acoustic resonance through its shape

TABLE II
DSP, ML, AND DSP+ML ON MICROCONTROLLER MEMORY USAGE SUMMARY

Category	Model	Flash Memory[a]	RAM
DSP Only	LM[b]	22.02 kB (8.88%)	2.84 kB (35.45%)
	LUT[c]	50.49 kB (20.36%)	2.84 kB (35.55%)
ML Only	SVM	43.08 kB (17.37%)	1.10 kB (13.70%)
	RF	134.09 kB (54.07%)	1.10 kB (13.73%)
DSP + ML	SVM+LM[b]	65.10 kB (26.25%)	3.93 kB (49.15%)
	SVM+LUT[c]	93.57 kB (37.73%)	3.94 kB (49.24%)
	RF+LM[b]	156.12 kB (62.95%)	3.93 kB (49.18%)
	RF+LUT[c]	184.61 kB (74.44%)	3.94 kB (49.28%)

[a] Flash Memory of the ATmega 2560 (248 kB Flash, 8 kB RAM available).
[b] LM: Linear Model implementation in DSP.
[c] LUT: Lookup Table implementation in DSP.

that focuses and amplifies sound waves. A Raspberry Pi 3B+ is connected to the Mayfly via USB to trigger recording and logging weather data from a local API. A nearby Google Nest camera provides a live feed of the deployment site and weather conditions for data labeling (ground truth).

C. RF Model Training and Evaluation

Features are extracted using MFCCs and labeled by precipitation category. Environmental audio files with 83 samples for Mixed Precip., 62 samples for rain, 101 samples for snow, and 70 samples for WGN [see Table ST3 in the supplementary material] are utilized in training. Leave-one-out cross-validation (LOO-CV) is used for verification, where one sample is held for testing, and the remaining samples are for training via `fit`/`predict`. The training is carried out on a computer, and the model is ported as a header file and included in the Arduino program. The RF classifier achieves 97.35% precision in simulated and 85.67% in recorded environmental samples. Class-wise performance is shown with confusion matrices and confidence plots in Figs. S9 and S10, respectively.

D. SVM Model Training and Evaluation

The SVM classifier is similarly trained on MFCC-derived features using a linear kernel and evaluated with LOO-CV. Accuracy reaches 98.07% on synthetic (simulated) data and 85.99% on real environmental recordings. Confusion matrices and class-wise confidence histograms visualize model performance (Figs. S9 and S10).

E. ML on the AURA Board

The SVM shows the lowest memory footprint among ML models and higher accuracy than RF (Table II and Fig. S9). SVM+LM uses 65.10 kB (26.25%), the smallest overall, followed by SVM+LUT at 93.57 kB (37.73%). RF+LM and RF+LUT require 156.12 kB (62.95%) and 184.61 kB (74.44%), respectively, with a combined (ML+DSP) RAM usage of approximately 3.93–3.94 kB (≈ 49%). Fig. S9 in the supplementary document shows the confusion matrices for the SVM and RF.

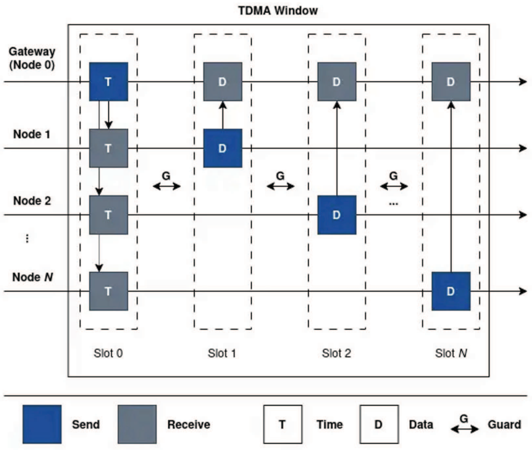

Fig. 8. Designed TDMA protocol for the AURA LoRa network.

VI. IoT Sensor Network

This section describes the LPWAN development. The LoRa network and packet format details, in addition to the TDMA protocol, are discussed. A multinodal deployment representation is provided.

A. LoRa Star Network

The LoRa network is based on a star topology with AURA nodes transmitting to a central Iridium-enabled gateway [39], [40], [41], [42]. Leaf Nodes use RFM95 modules supporting up to 15-km line-of-sight range [43], with experimental validation of up to 1.23 km in urban conditions with surrounding buildings and trees. LoRa enables low-power, LoRa communication.

B. TDMA Protocol

The TDMA protocol shown in Fig. 8 controls the wireless node communication by dividing time into fixed intervals, ensuring accurate data transmission while preventing LoRa packet collisions and minimizing power consumption [44]. Each node is assigned a unique time slot based on an auto-incrementing ID, starting from 0, with the gateway node always occupying ID 0. A brief guard period at the end of each interval prevents timing conflicts during transitions [see Algorithms 2 and 3 in the supplementary material for parameters in the TDMA Protocol section].

TDMA windows are initiated at some predetermined frequency and consist of two stages: 1) synchronization, where a sync packet shares the gateway's time, start of the next window, and per-node acknowledgments; and 2) data transfer, where nodes transmit in slots of the window determined based on their node ID [Algorithms 2 and 3 in the supplementary document]. Data packets are stored on the local SD card until their sequence number (timestamp) is acknowledged in the gateway's sync packet. During transmission windows, leaf nodes send as many stored packets as possible. Each packet's header contains the timestamp of the oldest unacknowledged packet and a transmission number [see in the supplementary document]. Only when the gateway receives a packet with a timestamp that has been marked as being waited on can it update its latest acknowledgment from the sending node. The transmission numbers within a series of data packet transmissions allow the gateway to detect any dropped packets during the send window. The gateway acknowledges the timestamp of a packet only if its transmission number is exactly one greater than the previous transmission number. Otherwise, the gateway assumes the packet has been dropped. We find that the parameters $S_s = 20$ (slot duration), $G_s = 1$ (guard period), and $D_s = 1$ (transmission delay) provide adequate time for peripherals to complete any tasks they are running, enter their sending window, and transmit any queued packets without sending outside of their slot.

C. LoRa Packet Structure

The sync packet (Fig. S13) is transmitted by the gateway at the start of each TDMA cycle to all nodes. It includes a 12-byte header and a payload containing two 4-byte Unix timestamps and one 4-byte acknowledgment entry per node. The data packet (also in Fig. S13), sent from leaf nodes to the gateway, begins with the same header followed by a 1-byte flag field, which indicates the presence of optional sensor data. Depending on the flags, the payload optionally includes a 1-byte precipitation class, temperature, and humidity stored in a 1-byte quantized format, and three sensor readings stored as raw 2-byte voltages. This structure enables efficient communication and heterogeneous node capabilities.

D. Iridium Satellite Uplink

A UART interface links the AURA gateway to a RockBLOCK 9603 modem with an Iridium 9603 transceiver. Sensor data gets buffered and uplinked via the Short-Burst Data (SBD) protocol, utilizing 340 bytes for uplink and 270 bytes for downlink. Messages are routed through the Iridium network to Cloudloop, then via webhook to a web app, which parses and saves them to a database. The details from the Cloudloop server front-end, the Iridium satellite connection, and the sensor network are provided in Figs. S11 and S12.

E. AURA Network Deployment Setting

The AURA system is designed for deployment in mountainous regions. A representative network configuration is shown in Fig. 9, where up to five devices are deployed, with one functioning as a satellite-enabled gateway. The gateway collects its precipitation data while aggregating data from the other leaf nodes and forwarding all information to the Cloudloop server via the satellite network in NRT. Each sensor node features a solar panel and battery.

Under harsh mountainous conditions and snow, limited solar irradiance during extended winter periods can hinder battery charging. To address this, we propose leveraging the temperature and humidity sensor only to estimate the dew point using $T_d = T - (100 - \mathrm{RH})/5$, where T_d is the dew point in Celsius, T is the ambient temperature in Celsius, and RH is the relative humidity in percent (%). The system applies a dew point threshold to opportunistically wake and record audio only when conditions suggest a high likelihood of precipitation.

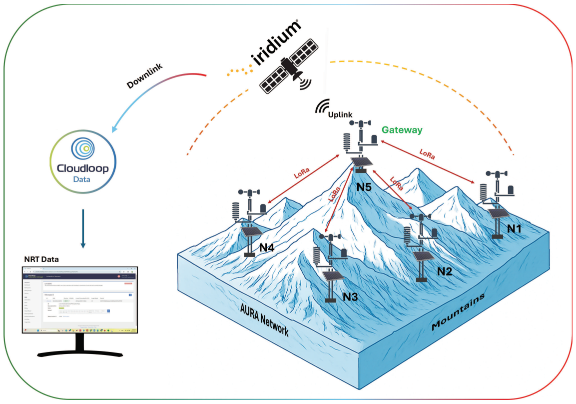

Fig. 9. IoT AURA sensor network deployment in a mountain setting scenario with LoRa and satellite communication.

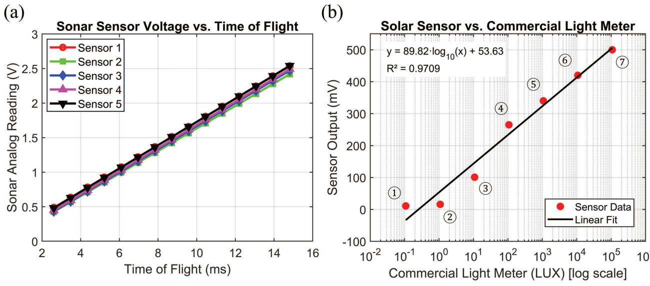

Fig. 10. AURA sensor calibration curves. (a) MB7060 sonar (snow depth) sensor calibration for 5 different sensors. (b) Solar sensor calibration in (1) full moon, (2) deep twilight, (3) twilight, (4) very dark day, (5) overcast day, (6) full daylight, and (7) direct sunlight.

VII. SENSOR CALIBRATION

Calibration of sensors is important for obtaining consistent and accurate results. This section details the calibration and validation of the meteorological sensors. The calibration results are illustrated in Fig. 10.

A. Sonar Sensor (Snow Depth)

The MB7060 sensors are placed and moved at known distances (d) in 15-cm increments ranging from 30 cm to 2.55 m, and the corresponding sensing voltages are recorded. At 22 ° C, the speed of sound (SoS) is 344.632 m/s, calculated using $\mathrm{SoS}(T) = 331.3 + 0.606T$, where T is the ambient temperature in Celsius. The discrete time-of-flight calculations are then determined by $\mathrm{ToF} = (2d)/(\mathrm{SoS})$. During calibration (Fig. 10(a), Table ST2), linear regressions are performed to relate sensor voltage to time-of-flight. The fit provides $U(v)$, a dynamic version of ToF as a function of sensor voltage v. In operation, $U(v)$ is used to compute ToF from real-time voltage readings. Snow depth is then calculated using the known ground distance, the estimated ToF, and the temperature-dependent SoS using $D = G - ((U(v))/2) \cdot \mathrm{SoS}(T)$, where G is the ground distance from the sensor.

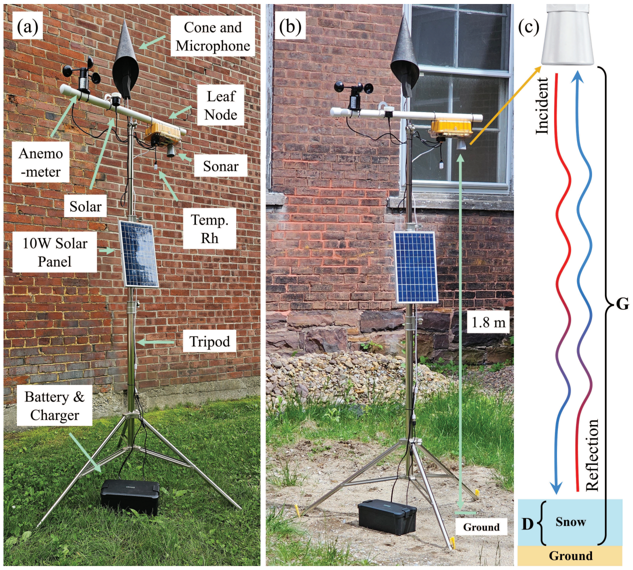

Fig. 11. AURA sensor system (single leaf node) field deployment. (a) System components and sensors, (b) deployment site, and (c) snow depth calculation methodology.

B. Illuminance (Solar Radiation) Sensor

The light sensor and its voltage are calibrated against an MT-912 light meter under seven different light conditions, as illustrated in Fig. 10(b). The results indicate linearity.

C. Temperature and Relative Humidity Sensor

Each DHT22 sensor is factory calibrated for both temperature and relative humidity in a precise calibration chamber. The calibration coefficients, which include temperature compensation, are stored in the sensor's one-time programmable (OTP) memory and automatically referenced during operation.

VIII. FIELD DEPLOYMENT

This section thoroughly explains the field deployment setup of a single sensor node to verify the operation, including ML, sensors, and prediction.

A. Practical Deployment Setup

Five AURA boards are developed and mounted in Pelican cases (Boxes#1–5); one setup for training data collection, one as a gateway, and three as leaf nodes (Box#1 is deployed in this study as the leaf node). Each unit integrates environmental sensors and a microphone with a conical chamber, mounted on a 1.8 m tripod [Fig. 11(a) and (b)]. Power is supplied via a 12-V and 12-Ah battery and solar panel with a charge controller in a waterproof enclosure. To mitigate wind noise, we utilize microphone foam covers (windscreen) in addition to mounting the microphone near the vertex of the conical resonance chamber. The sensor box contains the AURA board, wiring, and sensors, as shown in Fig. S8. The leaf node deployed in this section only features a single microphone.

B. Deployment Site

The sensor leaf node was deployed on the UVM campus in Burlington, Vermont, in May 2025, with a measurement interval set to every 10 min. The deployment area is surrounded by buildings to provide a controlled test bed for tripod stability tests and system verification.

IX. RESULTS AND DISCUSSION

This section presents the data collected using the AURA board, along with a comparison and analysis.

A. Field Data

The AURA system was deployed for 66.5 h (2.77 days) in a controlled environment to validate its performance. Field data collected from the Sonar sensor show a stable distance measurement averaging approximately 1.75 m, 2.78% lower than the reference value of 1.8 m. No frozen form of precipitation (i.e., hail or snow) was observed during the deployment course. Hence, only the ground distance is reported. Slight variations in sonar readings are caused by wind and minor displacements of the sensor. The solar sensor effectively captured fluctuations in sunlight during the day, with readings reflecting overcast and direct sunlight radiation due to quickly varying conditions, as illustrated in Fig. 12(a). Solar radiation is the main energy source driving snow melt, especially during clear and sunny days. Solar radiation measurement can be used to quantify snow melt rates and predict the timing of water runoff, which is critical for water resource management and flood forecasting. SVM-based precipitation classification during the deployment, shown in Fig. 12(b), identified three weather states: no precipitation (WGN), rain, and mixed precipitation (a combination of rain, hail, and snow precipitation). These predictions align with the recorded and visually observed events, supporting the viability of on-device ML. In Fig. 12(e), the historical data from Burlington weather are provided. Comparing our acoustic-based detection with the weather data, precipitation is detected between May 22nd and May 24th, which matches the historic weather data. Environmental measurements of temperature and relative humidity followed expected diurnal trends, as shown in Fig. 12(c), while wind velocity patterns were recorded in Fig. 12(d). Temperature trends follow the provided historic data [faint yellow line in Fig. 12(e)]. The deployment site conditions obstructed the wind and reduced the wind speed; hence, limited wind data was captured.

X. DISCUSSION OF PREDICTION AND OBSERVATION MISMATCH

Prediction errors and mismatches can be explained by three factors. First, the ground truth weather station is not collocated with the deployed system, thus the precipitation recorded at the station may not occur at the sensor site due to spatial variability. Second, the aluminum cone produced weak acoustic responses to light rain, as its rigid walls dampened small droplet impacts. A redesigned pyramid structure with transparent, flexible sheets is proposed to improve sensitivity. Third, the model was trained on a limited dataset, reducing generalization. Ongoing work includes collecting additional verified events using a Parsivel^2 disdrometer.

For classification [Fig. 12(f)], the model employs two labels: class **1** for precipitation and class **3** for no precipitation. A prediction of class **1** is correct when the station reports rain

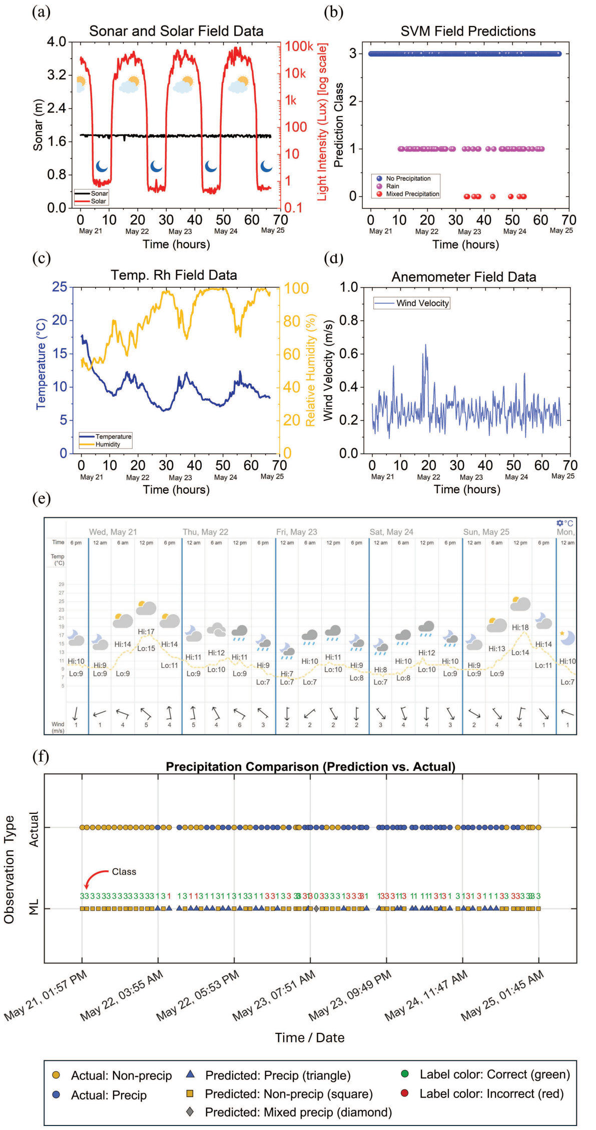

Fig. 12. Deployment data and system performance verification, (a) sonar and solar sensors data, (b) SVM ML predictions, (c) temperature and relative humidity sensor readings, (d) wind velocity, (e) historic weather screenshot for Burlington, VT from timeanddate.com, and (f) observation type comparison timeline for the SVM model and local weather station; for panel interpretation, refer to legend and color formatting.

events (rain, light rain, or showers), and class **3** is correct under non-precipitation conditions (clear, cloudy, or overcast). Data collected by AURA is reprocessed to adjust the time stamps. In total, 72.73% of events are correctly detected. Due to spatial variability and distance from the weather station, the comparison mainly serves as proof-of-concept.

A. *Comparing AURA With State-of-the-Art*

Compared to the literature, the proposed AURA system integrates multiple modalities and sensors into a compact wireless system that is verified to detect precipitation phase. Many systems only rely on single precipitation type detection as outlined in Table III. The AURA system integrates LoRa networking and satellite data transmission for NRT precipitation data transmission, supporting distributed deployments and NRT reporting in remote settings, unlike other systems.

TABLE III

SENSORS AND FEATURES COMPARISON: AURA VERSUS LITERATURE

Sensor / Feature	**AURA**	**[13]**	**[14]**	**[35]**
On-chip DSP	MFCCs	MFCCs	STFT	Sens. Read
On-chip ML	SVM	CNN	CNN+FNN	SVM
Technology	Acoustic	Acoustic	Acoustic	Meteorological
Snow Depth (cm)	30–400	N/A	N/A	N/A
Anemometer (m/s)	up to 32.4	N/A	N/A	✓
Temp. Range ($^\circ$C)	-40 to 125	N/A	N/A	✓
Sensors	up to 16	Mic.	N/A	Temp/Hum/Wind
SD Card	✓	✓	✓	N/A
LoRa (915 MHz)	✓	N/A	N/A	N/A
Satellite	Gateway ✓	N/A	N/A	N/A
Low Power	✓	✓	N/A	✓

XI. CONCLUSION

To conclude, we present the design and validation of the AURA system, which is a low-power, edge-based acoustic and meteorological distributed sensing platform for real-time precipitation phase classification. By integrating MFCC-based feature extraction with SVM and RF classifiers, the system achieves promising accuracies and precipitation detection. The incorporation of auxiliary sensors and LPWAN technologies enables robust, NRT data transmission from remote locations. The field deployment validates the performance of the entire system, and these results demonstrate the feasibility of embedded acoustic sensing for environmental monitoring, offering a scalable solution to improve hydrological and climate-related models.

Future work on AURA includes refining ML models, excluding noise-sensitive MFCCs, integrating weather sensors, and developing efficient LoRa multihop protocols. We are currently deploying our sensor instrumentation alongside a Parsivel^2 laser disdrometer to generate a long-term, high-quality dataset with accurate ground-truth labeling for future training and testing. We will use this data to improve model prediction accuracy, comparing SVM against RF and even a small neural network performance. We will consider other model input features available on the AURA platform, in addition to MFCCs, including time-domain acoustic features (e.g., zero-crossing rate), and nonacoustic environmental data such as temperature, relative humidity, and solar radiation. We also plan to evaluate various acoustic resonance chambers in real deployments and their impact on model performance. We anticipate that our system will support improved measurements in real and diverse environmental monitoring applications.

REFERENCES

[1] W. Xiong, G. Tang, T. Wang, Z. Ma, and W. Wan, “Evaluation of IMERG and ERA5 precipitation-phase partitioning on the global scale,” *Water*, vol. 14, no. 7, p. 1122, Mar. 2022.

[2] M. L. Wrzesien, T. M. Pavelsky, S. P. Sobolowski, L. S. Huning, J. S. Cohen, and J. D. Herman, “Tracking the impacts of precipitation phase changes through the hydrologic cycle in snowy regions: From precipitation to reservoir storage,” *Frontiers Earth Sci.*, vol. 10, pp. 1–21, Sep. 2022.

[3] G. Konapala, A. K. Mishra, Y. Wada, and M. E. Mann, “Climate change will affect global water availability through compounding changes in seasonal precipitation and evaporation,” *Nature Commun.*, vol. 11, no. 1, pp. 1–10, Jun. 2020.

[4] Y. Zhong et al., "Over 60% precipitation transformed into terrestrial water storage in global river basins from 2002 to 2021," *Commun. Earth Environ.*, vol. 6, no. 1, pp. 1–10, Jan. 2025.

[5] K. S. Jennings, T. S. Winchell, B. Livneh, and N. P. Molotch, "Spatial variation of the rain–snow temperature threshold across the Northern Hemisphere," *Nature Commun.*, vol. 9, no. 1, pp. 1–9, Mar. 2018.

[6] M. Fehlmann, M. Rohrer, A. von Lerber, and M. Stoffel, "Automated precipitation monitoring with the Thies disdrometer: Biases and ways for improvement," *Atmos. Meas. Techn.*, vol. 13, no. 9, pp. 4683–4698, Sep. 2020.

[7] H. Messer and O. Sendik, "A new approach to precipitation monitoring: A critical survey of existing technologies and challenges," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 110–122, May 2015.

[8] A. Bédard-Therrien et al., "Leveraging a radar-based disdrometer network to develop a probabilistic precipitation phase model in eastern Canada," *Hydrol. Earth Syst. Sci.*, vol. 29, no. 4, pp. 1135–1158, Feb. 2025.

[9] E. Adirosi et al., "Database of the Italian disdrometer network," *Earth Syst. Sci. Data*, vol. 15, no. 6, pp. 2417–2429, Jun. 2023, doi: 10.5194/essd-15-2417-2023.

[10] A. Trucco, R. Bozzano, E. Fava, S. Pensieri, A. Verri, and A. Barla, "A supervised learning approach for rainfall detection from underwater noise analysis," *IEEE J. Ocean. Eng.*, vol. 47, no. 1, pp. 213–225, Jan. 2022.

[11] G. Cui et al., "Detecting rain–snow-transition elevations in mountain basins using wireless sensor networks," *J. Hydrometeorol.*, vol. 21, no. 9, pp. 2061–2081, 2020.

[12] R. Kumari, D. K. Sah, K. Cengiz, N. Ivković, A. Gehlot, and B. Salah, "Acoustic signal-based indigenous real-time rainfall monitoring system for sustainable environment," *Sustain. Energy Technol. Assessments*, vol. 60, Dec. 2023, Art. no. 103398.

[13] R. Avanzato and F. Beritelli, "An innovative acoustic rain gauge based on convolutional neural networks," *Information*, vol. 11, no. 4, p. 183, Mar. 2020.

[14] J. L. E. Capin and E. D. Dimaunahan, "Implementation of neural network models using acoustic and spectral features for rainfall intensity classification," in *Proc. 4th Int. Conf. Ind. Eng. Artif. Intell. (IEAI)*, Apr. 2023, pp. 23–29.

[15] C. Mallary, C. J. Berg, J. R. Buck, and A. Tandon, "Listening for rain: Principal component analysis and linear discriminant analysis for broadband acoustic rainfall detection," *J. Acoust. Soc. Amer.*, vol. 154, no. 1, pp. 556–570, Jul. 2023.

[16] Ma. M. S. Pangaliman, F. R. G. Cruz, and T. M. Amado, "Machine learning predictive models for improved acoustic disdrometer," in *Proc. IEEE 10th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2018, pp. 1–5.

[17] M. Wang et al., "Estimating rainfall intensity based on surveillance audio and deep-learning," *Environ. Sci. Ecotechnology*, vol. 22, Nov. 2024, Art. no. 100450.

[18] X. Li, Y. He, A. Peng, and K. Yao, "Rainfall recognition based on multi-feature fusion of audio signals," in *Proc. 4th Int. Conf. Comput. Sci. Manage. Technol.*, Oct. 2023, pp. 338–344.

[19] N. J. Kinar and J. W. Pomeroy, "SAS2: The system for acoustic sensing of snow," *Hydrological Processes*, vol. 29, no. 18, pp. 4032–4050, Aug. 2015.

[20] Z. Kh. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122136–122158, 2022.

[21] M. Maayah, A. Abunada, K. Al-Janahi, M. E. Ahmed, and J. Qadir, "LimitAccess: On-device TinyML based robust speech recognition and age classification," *Discover Artif. Intell.*, vol. 3, no. 1, Feb. 2023, doi: 10.1007/s44163-023-00051-x.

[22] A. Andreadis, G. Giambene, and R. Zambon, "Monitoring illegal tree cutting through ultra-low-power smart IoT devices," *Sensors*, vol. 21, no. 22, p. 7593, Nov. 2021.

[23] E. Sasmaz and F. B. Tek, "Animal sound classification using a convolutional neural network," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sep. 2018, pp. 625–629.

[24] O. K. Toffa and M. Mignotte, "Environmental sound classification using local binary pattern and audio features collaboration," *IEEE Trans. Multimedia*, vol. 23, pp. 3978–3985, 2021.

[25] A. Qurthobi, R. Damaševičius, V. Barzdaitis, and R. Maskeliūnas, "Robust forest sound classification using Pareto-mordukhovich optimized MFCC in environmental monitoring," *IEEE Access*, vol. 13, pp. 20923–20944, 2025.

[26] M. Chen, X. Wang, M. Wang, X. Liu, Y. Wu, and X. Wang, "Estimating rainfall from surveillance audio based on parallel network with multi-scale fusion and attention mechanism," *Remote Sens.*, vol. 14, no. 22, p. 5750, Nov. 2022.

[27] M. Abdollahi, E. Henry, P. Giovenazzo, and T. H. Falk, "The importance of context awareness in acoustics-based automated beehive monitoring," *Appl. Sci.*, vol. 13, no. 1, p. 195, Dec. 2022.

[28] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Comput. Surveys*, vol. 54, no. 8, pp. 1–37, Nov. 2022.

[29] H. Hua et al., "Edge computing with artificial intelligence: A machine learning perspective," *ACM Comput. Surv.*, vol. 55, no. 9, pp. 1–35, 2023.

[30] P. Grzesik and D. Mrozek, "Combining machine learning and edge computing: Opportunities, challenges, platforms, frameworks, and use cases," *Electronics*, vol. 13, no. 3, p. 640, Feb. 2024.

[31] S. Caro-Via, E. Vidaña-Vila, G. J. Ginovart-Panisello, C. Martínez-Suquía, M. Freixes, and R. M. Alsina-Pagès, "Edge-computing meshed wireless acoustic sensor network for indoor sound monitoring," *Sensors*, vol. 22, no. 18, p. 7032, Sep. 2022.

[32] L. Hou et al., "Intelligent microsystem for sound event recognition in edge computing using end-to-end mesh networking," *Sensors*, vol. 23, no. 7, p. 3630, Mar. 2023.

[33] Q. B. Pham, E. Łupikasza, and M. Łukasz, "Classification of precipitation types in Poland using machine learning and threshold temperature methods," *Sci. Rep.*, vol. 13, no. 1, pp. 1–11, Nov. 2023.

[34] H. H. Qasim et al., "Enhancing weather monitoring: A comprehensive study utilizing IoT, ESP32, sensor integration, and blynk platform," in *Proc. IEEE 10th Int. Conf. Smart Instrum., Meas. Appl. (ICSIMA)*, Jul. 2024, pp. 156–161.

[35] W. P. Putra, R. Robiyanto, A. Puspaningrum, A. Rifai, R. Burjulius, and A. Sumarudin, "Energy-efficient rainfall prediction using support vector machine on edge ai platforms," *Int. J. Informat. Visualizat.*, vol. 8, nos. 3–2, pp. 1686–1692, Nov. 2024.

[36] C. K. Sajil et al., "Edge analytics enabled acoustic rain gauge," in *Proc. IEEE Recent Adv. Intell. Comput. Syst. (RAICS)*, Nov. 2024, pp. 1–5.

[37] N. Schizas, A. Karras, C. Karras, and S. Sioutas, "TinyML for ultra-low power AI and large scale IoT deployments: A systematic review," *Future Internet*, vol. 14, no. 12, p. 363, Dec. 2022.

[38] M. Alhlffee, "MFCC-based feature extraction model for long time period emotion speech using CNN," *Revue d'Intelligence Artificielle*, vol. 34, no. 2, pp. 117–123, May 2020.

[39] S. F. Hagh et al., "Autonomous UAV-mounted LoRaWAN system for real-time monitoring of harmful algal blooms (HABs) and water quality," *IEEE Sensors J.*, vol. 24, no. 7, pp. 11414–11424, Apr. 2024.

[40] S. F. Hagh et al., "A low-cost LoRA optical fluorometer–nephelometer for wireless monitoring of water quality parameters in real time," *IEEE Sensors J.*, vol. 24, no. 13, pp. 21511–21519, Jul. 2024.

[41] S. F. Hagh et al., "LoRa pore pressure probe for real-time coastal environment monitoring," *IEEE Sensors J.*, vol. 24, no. 23, pp. 39751–39758, Dec. 2024.

[42] P. Amngostar, M. Dehabadi, S. F. Hagh, A. R. Badireddy, D. Huston, and T. Xia, "LoRaStat: A low-cost portable LoRaWAN potentiostat for wireless and real-time monitoring of water quality and pollution," *IEEE Sensors J.*, vol. 25, no. 1, pp. 1561–1570, Jan. 2025.

[43] U. A. Ahmad et al., "Design of LoRA technology as GPS tracker and SoS panic button on fish lift nets," in *Proc. 6th Int. Semin. Res. Inf. Technol. Intell. Syst. (ISRITI)*, 2023, pp. 254–259.

[44] A. Bhawiyuga, "Data collision avoidance based on TDMA in LoRa wireless sensor networks," *J. Inf. Technol. Comput. Sci.*, vol. 9, no. 2, pp. 195–207, Aug. 2024.