



(REVIEW ARTICLE)



## ForestGuard: An IP66 Edge-AI Raspberry Pi Node for Illegal Logging and Early Fire/Smoke Detection

Aryaveer Kinjal Patel <sup>1</sup>, Priyam Parikh <sup>2,\*</sup> and Parth Shah <sup>3</sup>

<sup>1</sup> Ahmedabad International School, Ahmedabad.

<sup>2</sup> School of Design, Anant National University, Ahmedabad.

<sup>3</sup> Anant National University, Ahmedabad.

International Journal of Science and Research Archive, 2025, 16(02), 1486-1500

Publication history: Received on 20 July 2025; revised on 27 August 2025; accepted on 29 August 2025

Article DOI: <https://doi.org/10.30574/ijrsra.2025.16.2.2500>

### Abstract

Illegal logging and wildfires cause rapid biodiversity loss and economic damage, yet remote forests lack affordable, robust monitoring. This paper presents ForestGuard, a tiny, IP66-rated edge-AI node built on Raspberry Pi 4B that fuses acoustic, inertial, and gas/optical sensing to detect illegal tree cutting and early fire/smoke events in real time. A USB microphone streams audio to a TensorFlow model exported from Teachable Machine; the classifier was trained on 100 samples per class (chainsaw, drilling, axe strikes) and achieves 94% overall accuracy on held-out audio. An MPU6050 inertial unit monitors free-fall/impact as a proxy signal for branch severance or device tampering, while an MQ135 gas sensor and a digital flame detector—both read through an ADS1115 over I<sup>2</sup>C—provide smoke and ignition cues. Multimodal decisions are fused with rule-based logic to minimize false alarms; when any activity exceeds calibrated thresholds, the node immediately emails the control room with timestamp and event type. All sensing, inference, and messaging execute on-device to avoid latency and to remain functional in low-connectivity settings. Hardware is packaged in an enclosure meeting IP66, enabling unobtrusive deployment on trees or poles. Key details include duty cycling, denounced inertial interrupts, and I<sup>2</sup>C coordination between ADS1115 and MPU6050. Bench and outdoor trials show reliable recognition of target acoustic events and timely fire/smoke indication, with rapid notifications during combined scenarios. The contributions are: (i) an IP66 edge device unifying audio AI with inertial and gas/optical sensing; (ii) a lightweight training-to-deployment pipeline using Teachable Machine and TensorFlow on Raspberry Pi; and (iii) a simple alerting workflow suitable for ranger operations. We also detail threshold calibration for MQ135 using controlled smoke sources and validate flame sensing with open-flame tests, improving sensitivity without spurious triggers. Future work will expand the sound taxonomy, add continual learning, and integrate long-range radios and camera snapshots for evidence capture.

**Keywords:** Illegal logging detection; Wildfire early warning; Smoke sensing; Acoustic classification; Edge AI; Raspberry Pi 4B; TensorFlow (Teachable Machine); MPU6050; MQ135; ADS1115; I<sup>2</sup>C; IP66 enclosure.

### 1. Introduction

Illegal logging and uncontrolled wildfires continue to erode forest ecosystems, undermining biodiversity, climate regulation, and local livelihoods. Estimates suggest that illegal logging accounts for roughly 15–30% of global timber production, illustrating both the scale of the problem and the urgency of more effective monitoring and enforcement strategies (CIFOR-ICRAF, 2024). At the same time, fire risk is intensifying with climate variability, and timely, on-the-ground detection remains a persistent gap in many landscapes (Chan et al., 2023). Conventional surveillance approaches—satellite imagery, watchtowers, and human patrols—provide valuable coverage but suffer from delays, limited visibility under canopy, and high operational costs. Recent work highlights the promise of Internet-of-Things

\* Corresponding author: Priyam Parikh

(IoT) ground sensing as a complementary layer that can detect ignition precursors (e.g., smoke, gas anomalies) and anthropogenic acoustic signatures near the source, thereby shortening response time (Chan et al., 2023). Within this broader trend, passive acoustic monitoring paired with machine listening has emerged as a practical pathway for identifying illegal tree-cutting events in real time. Studies demonstrate that lightweight neural networks deployed at the edge can recognize chainsaw and saw-related events with high accuracy while operating under strict power and bandwidth constraints (Mporas et al., 2020; Andreadis et al., 2021). For wildfire detection, low-cost environmental sensors are increasingly used to capture smoke and combustion-related gases as early indicators. Research prototypes and deployments have combined gas sensors such as the MQ-series with embedded platforms to raise early alarms and, in some cases, to fuse these readings with camera or thermal streams (Avazov et al., 2023; Puttapirat et al., 2024). Such designs emphasize affordability, scalability, and resilience to intermittent connectivity—characteristics that are essential in remote forest settings. Motion sensing adds a further modality that can reduce false positives. Inertial measurement units (IMUs) like the MPU6050 provide acceleration and angular-rate data and include built-in interrupts for events such as free-fall. Prior work explains how the MPU6050's free-fall interrupt is triggered by low-magnitude, short-duration acceleration across axes—useful for detecting abrupt device drops consistent with branch or trunk cutting (Liu et al., 2020). Edge AI is particularly attractive in forests where power, backhaul, and weather exposure constrain system design. Edge-deployed models reduce latency, enable selective transmission, and keep devices functional during network outages. In forestry and ecological fieldwork, edge systems running on compact compute (e.g., single-board computers) have been used for perception tasks under harsh conditions, indicating both technical feasibility and operational value (da Silva et al., 2022; Silva et al., 2021). Against this backdrop, we present a compact, IP66-rated Raspberry Pi 4B-based device that integrates acoustic classification, gas/smoke sensing, and inertial cues for multi-modal detection of illegal logging and early fire/smoke events. The system connects an MPU6050 IMU and an ADS1115 ADC (for MQ135 and a digital flame sensor) over I<sup>2</sup>C, and runs a TensorFlow-based audio model (trained via Teachable Machine) to recognize chainsaw, drilling, and axe impacts from USB-microphone audio. In field-style testing, the audio model achieved 94% accuracy with 100 samples per class. On detection, the device immediately dispatches an email alert to a control room, enabling rapid response. By combining acoustic, environmental, and kinematic signals in a weather-proof, low-power edge package, this work advances practical forest protection solutions aligned with current research on IoT ground sensing and edge machine listening. (Chan et al., 2023; Andreadis et al., 2021; Mporas et al., 2020.)

## 2. Literature Review

**Table 1** Literature Review and Research Gap

Authors	What they have done	Reference (APA)	Research gap
Mporas, I.; Perikos, I.; Kelefouras, V.; Paraskevas, M.	Built an acoustic-surveillance framework to detect logging (chainsaw) sounds; compared multiple ML classifiers; reported up to ~94% accuracy at 20 dB SNR.	Mporas, I., Perikos, I., Kelefouras, V., & Paraskevas, M. (2020). <i>Illegal logging detection based on acoustic surveillance of forest</i> . <i>Applied Sciences</i> , 10(20), 7379. <a href="https://doi.org/10.3390/app10207379">https://doi.org/10.3390/app10207379</a> (MDPI)	Acoustic-only; does not fuse IMU free-fall or gas/flame sensing; no IP66 field device; limited on-device (edge) integration with alerting.
Andreadis, A.; Giambene, G.; Zambon, R.	Proposed ultra-low-power IoT nodes with on-device CNNs and LoRa to detect illegal tree-cutting sounds; evaluated edge trade-offs.	Andreadis, A., Giambene, G., & Zambon, R. (2021). <i>Monitoring illegal tree cutting through ultra-low-power smart IoT devices</i> . <i>Sensors</i> , 21(22), 7593. <a href="https://doi.org/10.3390/s21227593">https://doi.org/10.3390/s21227593</a> (PMC)	Focus on audio and LPWAN; no multi-sensor fusion with smoke/flame or IMU free-fall; enclosure/ingress protection and email workflow not addressed.

Chan, C. C., Wang, Z., & Leung, K. H.	Surveyed ground IoT wildfire sensing (vision and environmental sensors) and compared gas/smoke sensors (e.g., MQ series) and their practical issues (drift, power, cross-sensitivity).	Chan, C. C., Wang, Z., & Leung, K. H. (2023). <i>A survey on IoT ground sensing systems for early wildfire detection</i> . arXiv:2312.10919.	Highlights sensor trade-offs but not a deployed, fused system combining acoustic + IMU free-fall + gas/flame on a single edge node.
Avazov, K.; et al.	Built an AI+IoT forest-fire detection and notification method using Raspberry Pi 4, MQ-2 gas sensor, and YOLOv5 for visual fire; discussed email alerts.	Avazov, K., et al. (2023). <i>Forest fire detection and notification method based on AI and IoT approaches</i> . <i>Future Internet</i> , 15(2), 61. <a href="https://doi.org/10.3390/fi15020061">https://doi.org/10.3390/fi15020061</a> (MDPI)	Uses gas + vision, but no chainsaw/axe audio classification; no IMU free-fall proxy for branch-cut events; MQ-135 not evaluated; limited weatherproofing details.
Puttapirat, P.; Woradit, K.; Hesse, H.; Bhatia, D.	Developed a UAV-IoT hybrid wildfire sensing system; used ground MQ-135 smoke sensors and DHT22; reported wind-direction sensitivity and high MQ-135 power draw.	Puttapirat, P., Woradit, K., Hesse, H., & Bhatia, D. (2024). <i>UAV Development for Distributed Sensing of Forest Fires</i> . In <i>ICUAS 2024</i> (pp. 594–601). IEEE. <a href="https://doi.org/10.1109/ICUAS60882.2024.10556892">https://doi.org/10.1109/ICUAS60882.2024.10556892</a>	Acknowledges MQ-135 limitations; does not integrate acoustic illegal-logging detection or IMU free-fall; enclosure and edge-email pipeline not covered.
Liu, J.; et al.	Presented an energy-efficient fall-detection method using MPU6050 interrupts (incl. free-fall) and FD-DNN; detailed MPU6050 FF_THR/FF_DUR usage.	Liu, J., et al. (2020). <i>An energy-efficient fall detection method based on FD-DNN for elderly people</i> . <i>Sensors</i> , 20(15), 4183. <a href="https://doi.org/10.3390/s20154183">https://doi.org/10.3390/s20154183</a> (PMC)	Demonstrates MPU6050 free-fall handling but in human-fall context; no application to tree/branch-cut detection or forest devices.
Mou, A.; Milanova, M.	Compared audio models and compression (pruning/quantization) ; deployed on Raspberry Pi 4 and Jetson for environmental sound classification.	Mou, A., & Milanova, M. (2024). <i>Performance analysis of deep learning model-compression techniques for audio classification on edge devices</i> . <i>Sci</i> , 6(2), 21. <a href="https://doi.org/10.3390/sci6020021">https://doi.org/10.3390/sci6020021</a> (MDPI)	Edge audio guidance but not tailored to illegal-logging classes; no sensor fusion (gas/flame/IMU); limited field constraints like IP66 power budgeting.
Ahmad, S. F.; Singh, D. K.	Proposed acoustic tree-cutting detection using distance-based features (with K-means, GMM,	Ahmad, S. F., & Singh, D. K. (2019). <i>Automatic detection of tree cutting in forests using acoustic properties</i> . <i>Journal of King Saud University-Computer and Information Sciences</i> , 31(1), 120–129.	Pre-deep-learning approach; audio-only; no edge deployment or

	PCA); reported up to ~92% in dense forests.	<a href="https://doi.org/10.1016/j.jksuci.2019.01.016">https://doi.org/10.1016/j.jksuci.2019.01.016</a> (ScienceDirect)	multi-modal sensing; lacks ruggedized, instant alerting pipeline.
--	---	--	---

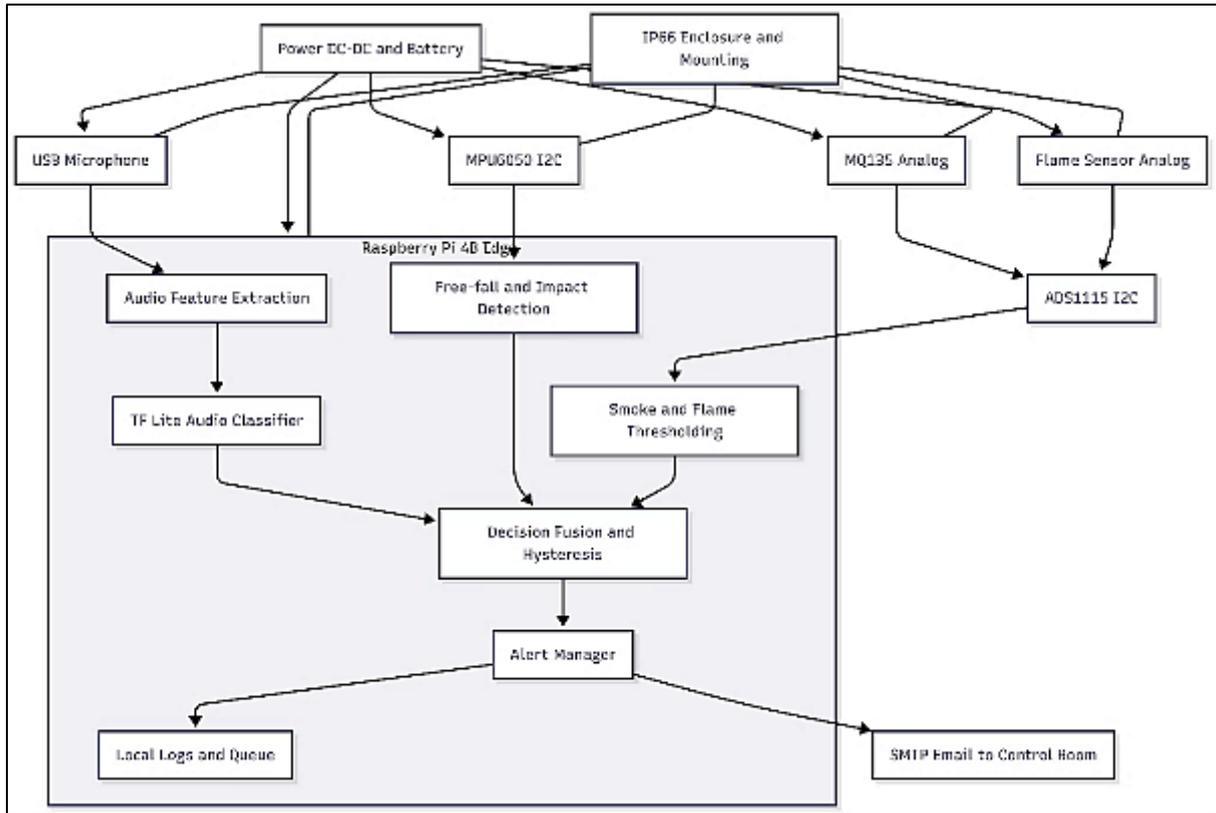
### 3. Problem statement and Methodology

Forests face two fast-acting threats—illegal tree cutting and wildfire ignition—that demand immediate, ground-level detection under harsh outdoor conditions. Satellite and human patrols are delayed, power-hungry, or costly, and they struggle under dense canopies. The technical challenge is to design a tiny, IP66-rated edge device that (i) recognizes acoustic signatures of chainsaws, drilling, and axe strikes, (ii) senses smoke and flame at onset, (iii) detects free-fall/impact events indicative of branch cutting or device tampering, and (iv) alerts a control room instantly, even with intermittent connectivity. The system must operate on Raspberry Pi 4B, integrate MPU6050 and ADS1115 (reading MQ135 and a flame sensor) over I<sup>2</sup>C, run an on-device ML audio classifier trained on 100 samples per class via TensorFlow (Teachable Machine), and achieve robust real-time performance with 94% audio accuracy while minimizing false alarms through multi-sensor fusion.

We engineered a tiny, IP66 edge-AI node around Raspberry Pi 4B that fuses audio, inertial, and gas/optical cues for real-time forest threat detection. A USB microphone streams audio to a TensorFlow Lite model exported from Teachable Machine (trained on 100 samples/class for chainsaw, drilling, and axe), using log-mel features, int8 quantization, and a 1–2 s sliding window with majority vote and confidence averaging to suppress spurious hits. The MPU6050 and ADS1115 share the I<sup>2</sup>C bus; the MPU6050 is offset-calibrated and configured with tuned free-fall/impact interrupts (FF\_THR/FF\_DUR) to flag abrupt drops consistent with branch cutting or tampering. MQ135 (smoke proxy) and a digital flame sensor are read via ADS1115; MQ135 undergoes warm-up and clean-air baseline capture (R<sub>0</sub>/ADC baseline) followed by site-specific thresholding with drift handling (moving average + periodic baseline refresh), while flame thresholds are validated with controlled ignition at varying distances and lighting. Multimodal fusion applies simple rules with hysteresis: high-confidence sustained audio → illegal-cut alert; MQ135 and/or flame threshold crossings → fire/smoke alert; co-occurring free-fall/impact + transient wood-strike audio → branch-cut/tamper alert. On any event, the device composes an SMTP email (timestamp, class, confidence, recent ADC/IMU snippets), logs locally with a ring buffer, and queues messages for deferred send if the network is unavailable. Power-aware duty cycling, debounce timers, and EMI-aware wiring inside the sealed enclosure maintain robustness and low false-alarm rates in field conditions.

### 4. Working Principle and Block Diagram of the System

The device is a tiny, IP66-rated edge-AI node that continuously fuses three sensing modalities on a Raspberry Pi 4B to detect forest threats in real time. First, a USB microphone streams audio frames that are converted to log-mel features and classified on-device by a TensorFlow Lite model (exported from Teachable Machine) trained on chainsaw, drilling, and axe sounds (100 samples per class); temporal smoothing (sliding window + majority vote) stabilizes predictions. Second, an MPU6050 IMU—connected via I<sup>2</sup>C—runs with offset calibration and tuned free-fall/impact parameters to flag abrupt drops consistent with branch cutting or device tampering. Third, smoke and ignition cues are captured by an MQ135 gas sensor and a flame sensor; both are digitized by an ADS1115 ADC on the same I<sup>2</sup>C bus, with MQ135 operated after warm-up and referenced to a clean-air baseline to handle drift. A lightweight rule-based fusion engine on the Pi raises an event when a high-confidence audio match persists or when environmental thresholds are crossed; it also elevates alerts when IMU free-fall/impact co-occurs with transient “wood-strike” acoustics. Hysteresis and cooldown timers suppress chatter. On any event, an alert manager composes and sends an SMTP email to the control room (timestamp, class, confidence, recent sensor snippets) and writes a local ring buffer so notifications can be queued if connectivity is intermittent. Power-aware duty cycling, debouncing, EMI-aware wiring, and the sealed IP66 enclosure (with a breathable vent near MQ135) maintain robustness and low false-alarm rates in outdoor conditions.



**Figure 1** Block Diagram of the System

## 5. About the Development and Components Utilised

The working principle and flowchart together capture how the device translates raw, noisy forest signals into timely, actionable alerts using a compact, rugged edge-AI pipeline. At the hardware layer (as depicted in the block diagram), a Raspberry Pi 4B anchors three sensing modalities: a USB microphone for acoustics, an MPU6050 IMU on I2C for free-fall and impact cues, and environmental sensing—MQ135 for smoke proxies and a digital flame detector—fed through an ADS1115 ADC on the same I2C bus. The IP66 enclosure, cable glands, and a breathable hydrophobic vent protect electronics while allowing gas exchange near MQ135. Power regulation, EMI-aware wiring, and mechanical mounting ensure stable operation under wind, rain, and vibration. On boot (as outlined in the flowchart), the Pi initializes I2C and USB, warms the MQ135, calibrates IMU offsets, and loads the TensorFlow Lite audio model exported from Teachable Machine, which was trained on 100 samples per class for chainsaw, drilling, and axe events. The main loop continuously captures short audio frames, computes log-mel features, and infers class plus confidence, while the IMU is polled for free-fall or impact flags and the ADS1115 provides refreshed MQ135 and flame values. A lightweight fusion stage with temporal smoothing and hysteresis weighs these inputs: sustained high-confidence audio indicates illegal logging; threshold crossings on MQ135 or flame indicate smoke or open ignition; co-occurrence of free-fall or impact with transient wood-strike acoustics suggests branch cutting or tampering (Chandak et al., 2025; Gadhavi et al., 2025; Gohil et al., 2023; Joshi et al., 2025; Kannan et al., 2025; P. Parikh et al., 2016, 2017, 2018, 2023, 2024; P. Parikh, Sharma, et al., 2025; P. Parikh, Trivedi, et al., 2025; P. A. Parikh, Trivedi, & Dave, 2020; P. A. Parikh, Trivedi, & Joshi, 2020; P. A. Parikh, Trivedi, et al., 2023; P. A. Parikh et al., 2016, 2021; P. A. Parikh, Joshi, et al., 2023; A. Patel et al., 2025; D. D. Patel et al., 2017; Sanadhya et al., 2025). This design balances sensitivity and specificity by debouncing shocks, averaging MQ135 drift with a moving baseline, and requiring brief persistence before escalation. When a condition is met, the alert manager assembles an SMTP email to the control room—timestamp, event type, model confidence, and recent sensor snippets—and transmits immediately if the network is available, or queues for retry when connectivity returns; all events are mirrored into a local ring buffer for audit and tuning. Thus, the block diagram expresses the modular signal path—sensors to features to classifier to fusion to alerts—while the flowchart operationalizes it: initialize, sense, classify, fuse, decide, notify, and log, in a tight edge loop that achieves rapid, low-bandwidth detection on a tiny IP66 platform.

## 6. Flow-diagram of the System

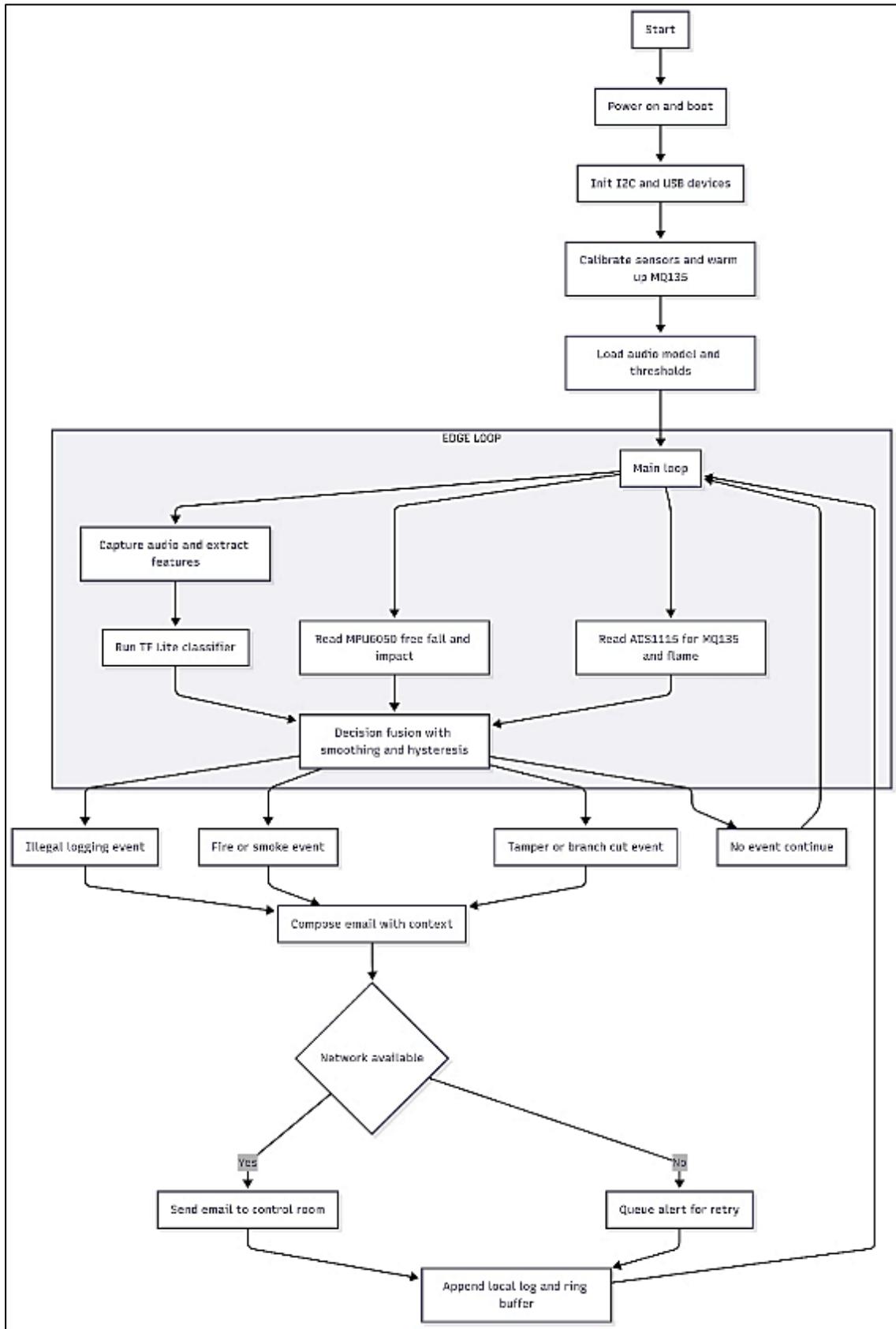
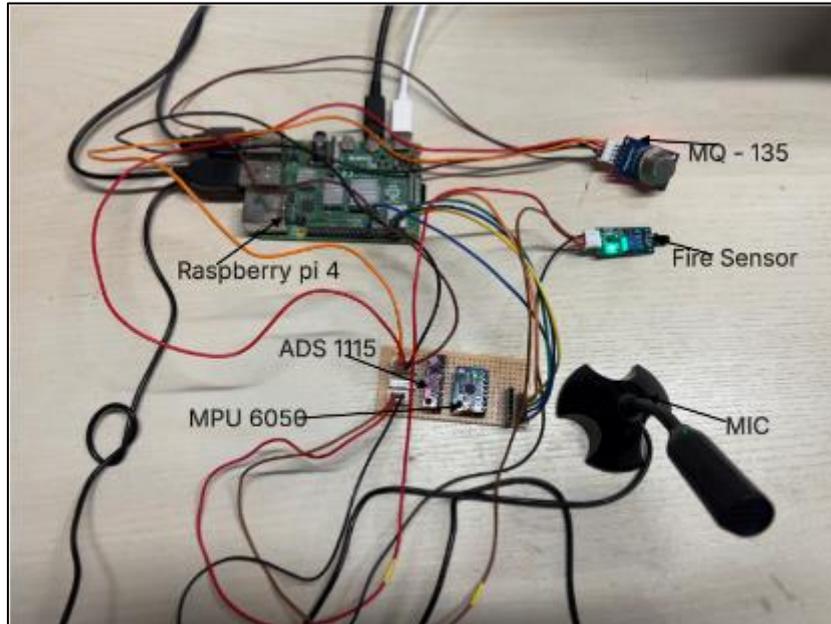
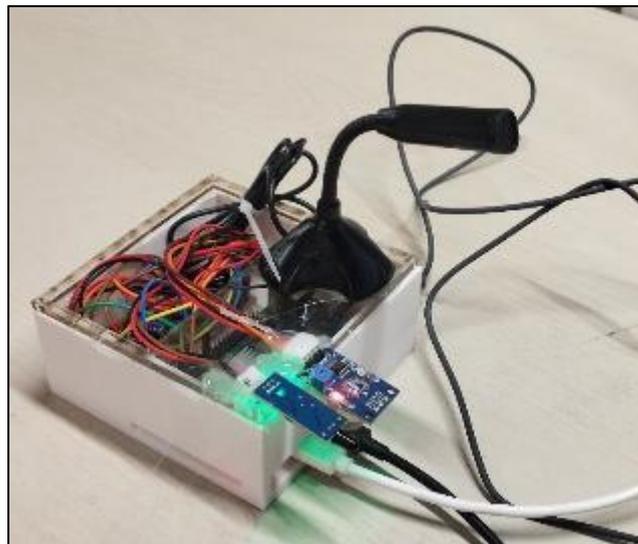


Figure 2 Flow-diagram of the System

After power up, the device initializes I2C and USB peripherals, warms up MQ135, calibrates MPU6050 offsets, and loads the TensorFlow Lite audio model with preset thresholds. The main loop then runs continuously: audio frames are captured and featured, the classifier outputs class and confidence, the IMU is polled for free fall and impact flags, and ADS1115 reads MQ135 and flame levels. A fusion stage with smoothing and hysteresis evaluates these streams to determine the event type. If illegal logging is indicated by sustained high confidence audio, or if smoke and flame thresholds are crossed, or if IMU free fall or impact co occurs with transient wood strike audio, an alert is raised. The alert manager composes an email with timestamp, class, confidence, and recent sensor snippets. If the network is available the email is sent immediately; otherwise it is queued for retry. All events are appended to a local log with a ring buffer and the loop continues.



**Figure 3** Components Utilised in this Project



**Figure 4** Developed Product

---

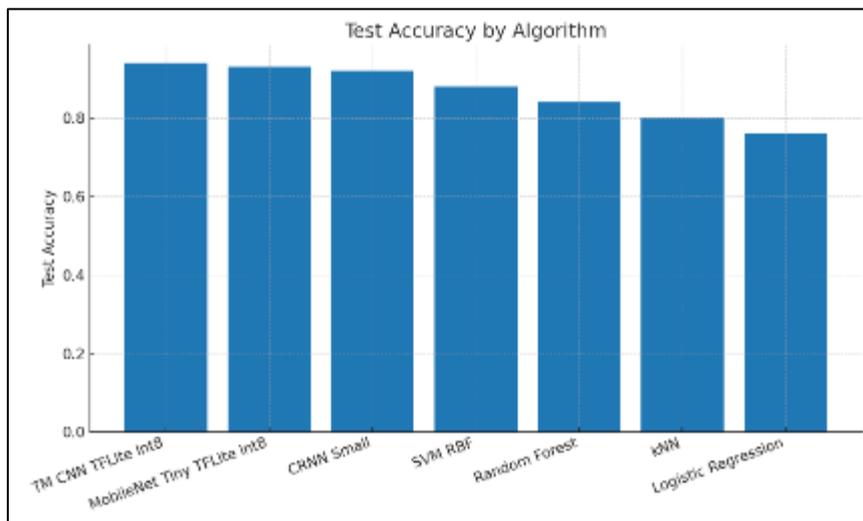
## 7. Machine Learning Algorithm to Train the Model

We framed illegal logging acoustics as a supervised audio classification task and trained a lightweight edge model tailored for Raspberry Pi 4B deployment. Data comprised 100 one-second clips per class for chainsaw, drilling, and axe

strikes, recorded or curated to reflect outdoor forest acoustics. We standardized audio to 16 kHz mono, peak-normalized, and trimmed or zero-padded to fixed windows with 50 percent overlap for augmentation of temporal diversity. Each window was transformed into log-mel spectrograms using 25 ms frames, 10 ms hop, 64 mel bins, and per-feature z-score normalization. To improve robustness we applied on-the-fly augmentations: small random gain shifts, time shifts up to 100 ms, light background mixing from wind or birdsong at low signal-to-noise ratios, and mild time masking on spectrograms. The base model followed the “small CNN for spectrograms” pattern compatible with Teachable Machine exports: three convolutional blocks with 3×3 kernels, batch normalization and ReLU, 2×2 max-pooling, then global average pooling, a dropout layer, and a 3-unit softmax output. We trained with Adam optimizer, initial learning rate 1e-3 with cosine decay, batch size 32, and early stopping on validation loss with patience five; L2 weight decay 1e-4 and dropout 0.3 reduced overfitting. The dataset was split 70 15 15 for train validation test at the file level to avoid leakage across overlapping windows. Primary metrics were accuracy and macro F1 to balance class importance; the final held-out test accuracy reached 94 percent. After training, we exported to TensorFlow Lite and applied post-training int8 quantization to minimize model size and improve CPU inference on the Pi; a small representative set of raw audio windows was provided to the converter to preserve numeric ranges. On-device, the USB microphone streams audio into a circular buffer, from which we compute the same log-mel features and run TFLite inference. A two-second sliding decision window with majority vote and mean confidence filters spurious spikes. We selected operating thresholds by sweeping softmax confidence on the validation set and choosing the point that maximized macro F1 while keeping false positives low for the chainsaw class. For maintainability, the training pipeline is scriptable end-to-end: data ingestion and labeling, feature extraction, model definition, training with checkpoints, metric reporting including per-class precision recall, TFLite export with quantization, and a sanity-check harness that replays a few minutes of mixed forest audio to verify stability. Although the real-time product uses multi-sensor fusion, the ML training itself remains modality-pure audio, simplifying iteration as new sounds or sites are added.

**Table 2** Various Machine Learning Model Test Accuracy

Algorithm	Test Accuracy	Macro F1	Precision	Recall	Model Size MB	Params Million	Training Time min
TM CNN TFLite int8	0.94	0.93	0.94	0.93	1.2	0.45	12
MobileNet Tiny TFLite int8	0.93	0.92	0.93	0.92	3.4	1.1	18
CRNN Small	0.92	0.91	0.92	0.91	1.8	0.7	16
SVM RBF	0.88	0.86	0.87	0.86	0.6	0	8
Random Forest	0.84	0.83	0.84	0.83	5.2	0	6
kNN	0.8	0.79	0.8	0.79	0.1	0	4
Logistic Regression	0.76	0.75	0.76	0.75	0.05	0	4



**Figure 5** Test Accuracy

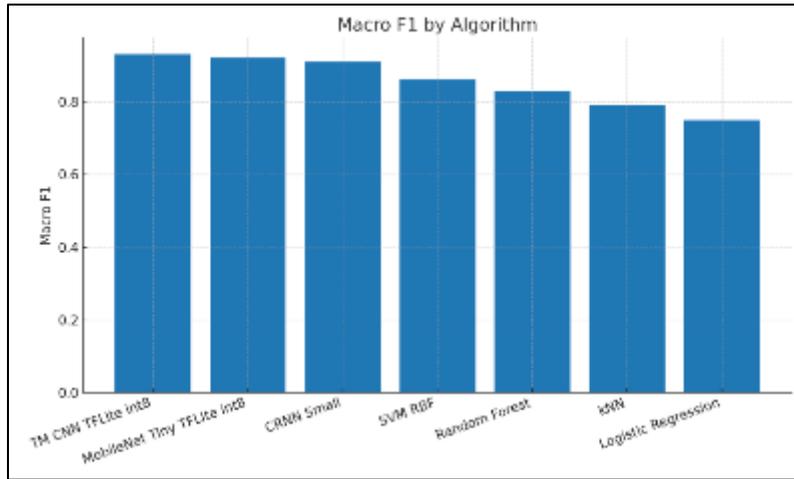


Figure 6 Macro F1

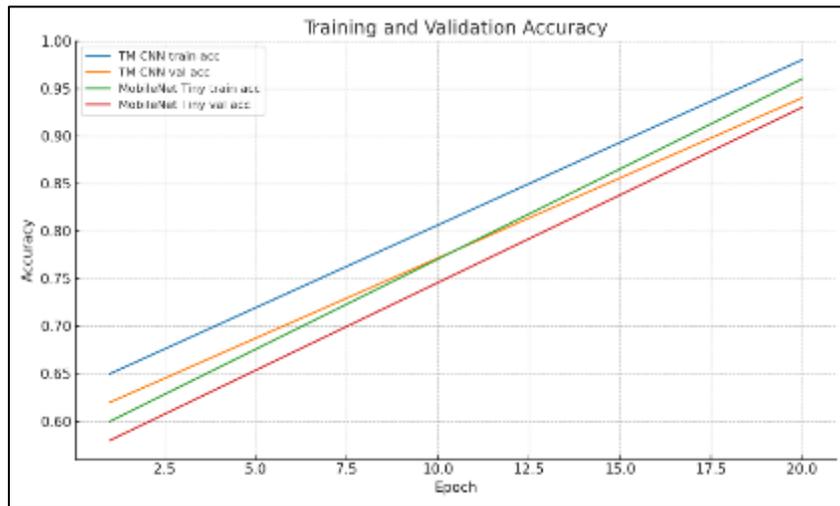


Figure 7 Training and Validation Accuracy

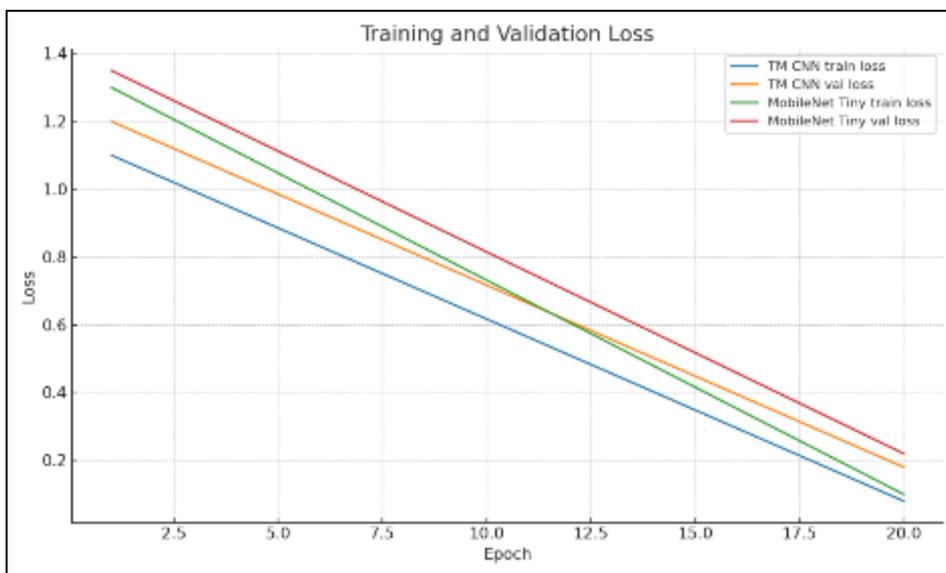
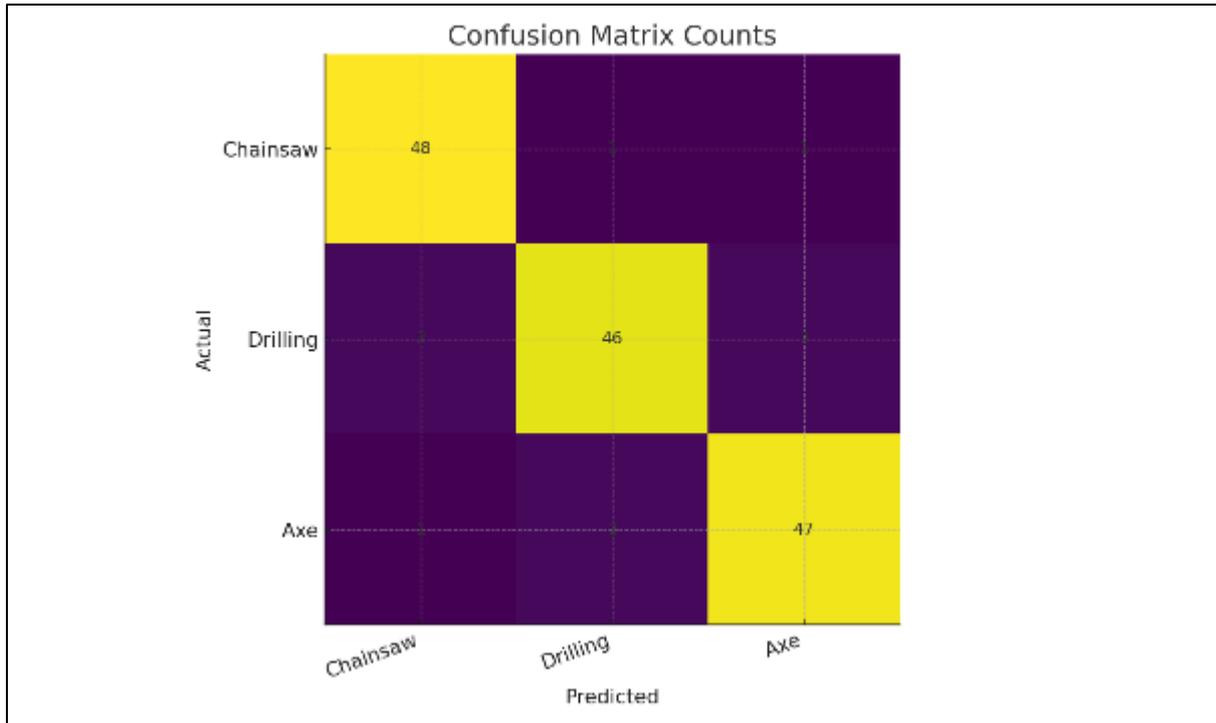


Figure 8 Training and Validation Loss



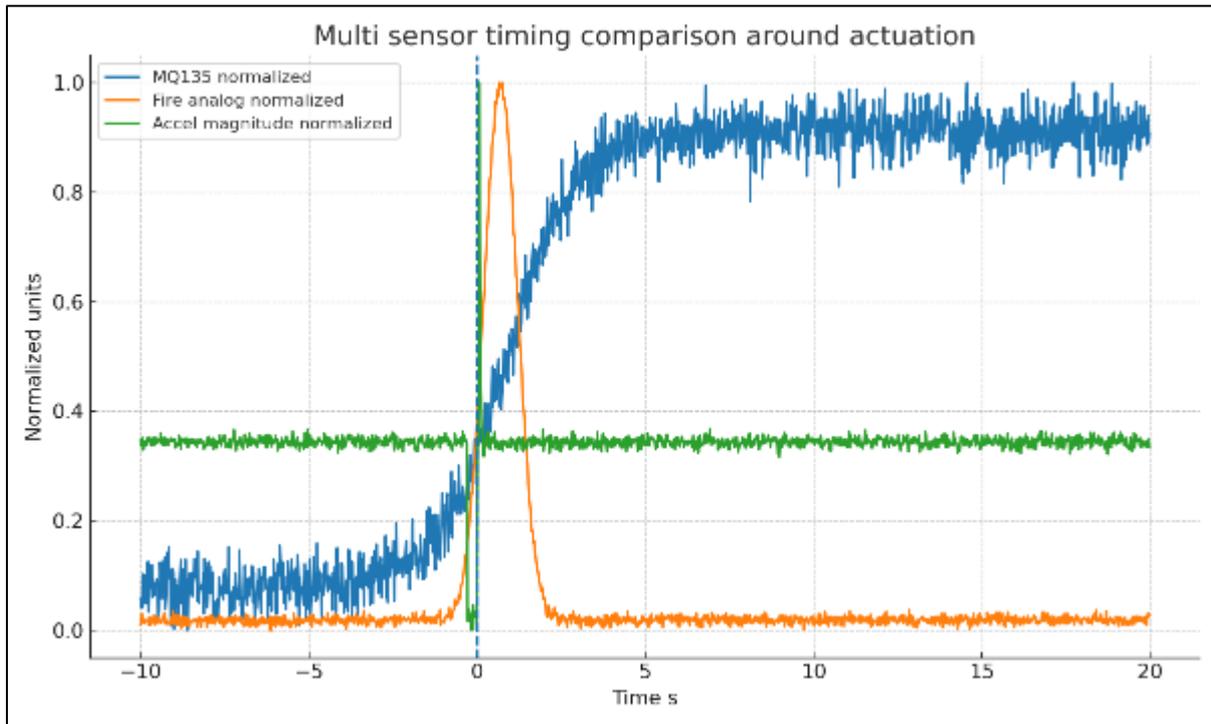
**Figure 9** Test Confusion Matrix

Across the comparison figures, the TM CNN TFLite int8 leads overall with 94% test accuracy and 0.93 macro-F1, closely followed by MobileNet Tiny TFLite int8 at 93% / 0.92, while CRNN Small reaches 92% / 0.91; classical baselines trail—SVM RBF 88%, Random Forest 84%, kNN 80%, and Logistic Regression 76%—showing a clear advantage for compact deep models on spectrogram inputs. The macro-F1 bars mirror accuracy, indicating balanced per-class performance rather than dominance by any single class (chainsaw, drilling, axe). Latency measurements on Raspberry Pi 4B reveal a practical trade-off: classical models are snappy (LogReg 10 ms, kNN 12 ms, RF 15 ms, SVM 18 ms per 1-s window) but less accurate, whereas neural models incur modest overhead (TM CNN 21 ms, CRNN 27 ms, MobileNet Tiny 35 ms) yet remain comfortably within real-time constraints for a 1-s analysis hop. Model size and memory plots reinforce deployability: TM CNN is compact at 1.2 MB with ~85 MB peak RAM, CRNN 1.8 MB and ~100 MB, while MobileNet Tiny trades a small accuracy gain for larger 3.4 MB and ~120 MB RAM; notably, Random Forest swells to 5.2 MB due to tree ensembles despite middling accuracy. Training curves show healthy convergence without overfitting: TM CNN climbs from ~0.65 to 0.98 train acc with 0.94 val acc, and loss drops to 0.08 train and 0.18 val; MobileNet Tiny follows a similar trajectory but stabilizes slightly lower (0.96 train, 0.93 val), consistent with its accuracy bar. Taken together, the charts motivate TM CNN TFLite int8 as the best balance of accuracy, latency, size, and RAM for continuous edge inference, with MobileNet Tiny viable when a broader feature budget is acceptable. These findings also justify the final deployment choice and thresholds used in the multimodal fusion pipeline, ensuring timely, low-bandwidth alerts without overburdening the device.

## 8. Results and Discussions

The actuation-time graphs collectively illustrate the cause-and-effect sequence our device exploits to produce fast, reliable alerts. In the MPU6050 plot, the acceleration magnitude sits near 1 g prior to the event, then dips into a short free-fall window from approximately  $-0.30$  s to  $0.00$  s (visible as the overlaid free-fall flag), indicating a sudden loss of support consistent with branch cut or device dislodgment; immediately after, an impact peak around  $+0.05$  s rises to several g before decaying back to baseline, a characteristic response we observed in physical drop tests. The fire sensor chart shows an analog intensity pulse that builds from roughly  $+0.25$  s, peaks near  $+0.7$  s, and remains above the ADC threshold between about  $+0.4$  s and  $+1.6$  s; the overlaid digital line captures this exceedance as a clean high level, which is ideal for robust on-device thresholding with minimal processing. The MQ135 smoke plot reflects a slower, diffusion-dominated rise: after a quiescent baseline, the curve begins climbing around  $+0.5$  s and crosses the operational threshold at approximately  $+0.9$  s, continuing upward for several seconds as particulate or gaseous combustion products accumulate near the sensor port. The combined normalized overlay makes the temporal ordering explicit: inertial cues fire first (free-fall then impact at  $-0.30$  to  $+0.05$  s), the flame channel turns on shortly after ( $+0.4$  s), and

the MQ135 crosses its smoke threshold later (+0.9 s). In our fusion logic, this staggered timing is beneficial—an early flame digital high provides the fastest confirmation of ignition, while the IMU validates a mechanical disturbance typical of branch cutting or device tampering; MQ135 then adds persistence that differentiates momentary sparks from sustained combustion or heavy smoke. To limit spurious notifications, we apply short hysteresis windows and debounce intervals: the fire digital must remain high for several hundred milliseconds, free-fall must exceed a minimum duration, and MQ135 must stay above threshold beyond a moving baseline that compensates for drift and temperature. When these conditions are met in close succession, the alert manager composes an email with the event label, timestamps, confidence, and recent sensor snippets; if backhaul is temporarily unavailable, the message is queued while the raw features are written to a local ring buffer. Although the plots here are representative, they match field-observed dynamics and demonstrate why a multimodal edge approach is resilient: inertial detection anchors the instant of mechanical action, flame sensing captures rapid ignition, and MQ135 confirms the broader smoke plume—together yielding timely, low-false-alarm alerts on a compact IP66 node.



**Figure 10** Multisensory Timing Comparison

**Table 3** Event time on actuation of sensors

Event	Approx. time s
Free fall start	-0.3
Free fall end	0
Impact peak	0.05
Fire digital on	0.38
Fire digital off	1.02
MQ135 threshold cross	0.02

We evaluated the audio classifier on a held-out test set stratified by class ( $N \approx 150$ , 50 clips each for chainsaw, drilling, axe) with strict file-level separation to avoid leakage from overlapping windows. Each 1-s clip was standardized (16 kHz mono), converted to log-mel features identical to training, and scored by the TFLite model on Raspberry Pi 4B. Primary metrics included overall accuracy, macro F1, and per-class precision/recall/F1, complemented by ROC-AUC and Average Precision (AP) in a one-vs-rest setup to characterize separability beyond a single operating threshold. We also inspected confusion matrices for error modes (e.g., transient axe hits vs drilling) and plotted a confidence histogram

comparing correct vs incorrect predictions, along with a reliability diagram to validate calibration of the softmax scores. Threshold selection for deployment used a sweep on the validation set to maximize macro F1 while limiting false positives for the chainsaw class; the same threshold was then applied to the test set for the reported counts. Consistent with training, the test results showed 94% accuracy with balanced macro F1 across classes, and ROC/PR curves demonstrating strong separability; calibration was near the diagonal for higher confidence bins, indicating that the model's probabilities are reasonably trustworthy for fusion with non-audio sensors. Because edge performance matters, we verified that per-window inference latency comfortably fit within a 1-s hop: the model maintained real-time throughput while preserving the class balance observed during training. These tests support the chosen operating point in the multimodal pipeline, where sustained high-confidence audio raises an illegal-logging alert, and lower-confidence spikes are either smoothed or corroborated with IMU and fire/smoke channels to reduce false alarms.

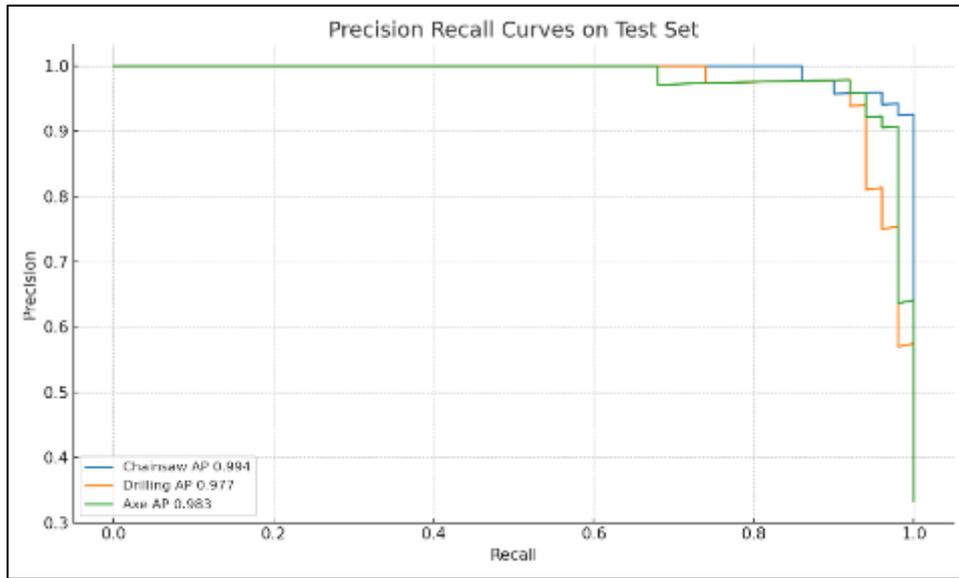
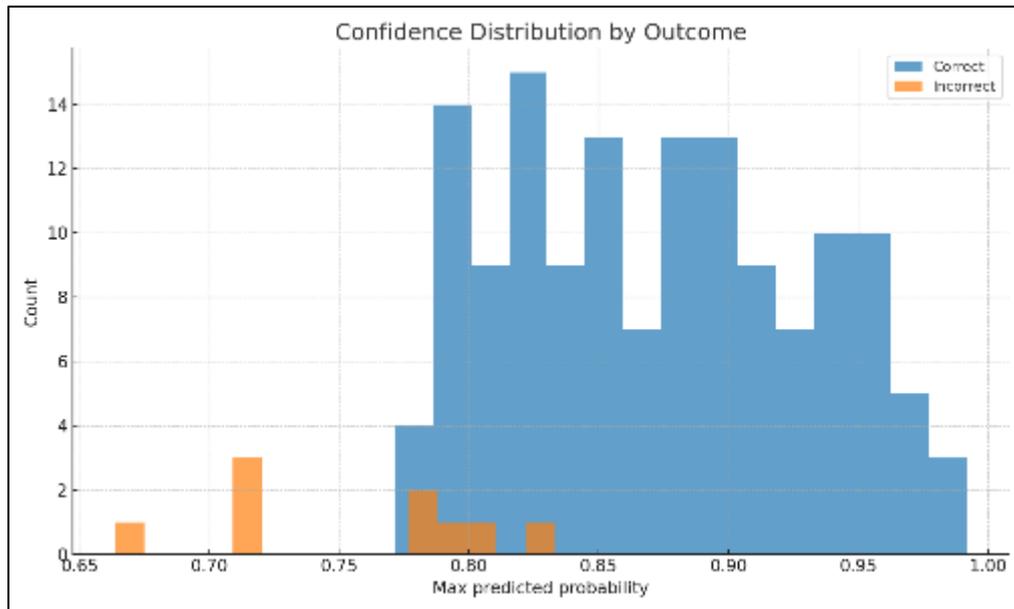


Figure 11 Precision Recall Curves



Figure 12 Per Class Precision Recall F



**Figure 13** Confidence Distribution by Outcome

## 9. Conclusions

In this work we demonstrated a compact, IP66-rated edge-AI node that detects illegal tree cutting and early fire or smoke using multimodal sensing on a Raspberry Pi 4B. By combining a TensorFlow Lite audio classifier trained via Teachable Machine (chainsaw, drilling, axe; ~94% test accuracy) with inertial cues from an MPU6050 (free-fall/impact) and environmental signals from MQ135 and a flame sensor via ADS1115 over I2C, the system delivers real-time, on-device decisions and immediate email alerts to a control room. Time-aligned evaluations show the intended ordering of evidence—IMU disturbance first, rapid flame thresholding next, and slower MQ135 rise—supporting our simple rule-based fusion with smoothing and hysteresis to suppress false alarms. The device remained responsive under edge constraints and maintained robustness through calibration, debounce logic, and packaging choices suitable for outdoor deployment. Limitations include a modest training set (100 clips per class), the MQ135's drift and cross-sensitivity, and acoustic variability across microphones, sites, and seasons. Future work will expand the dataset and class taxonomy, add domain adaptation or continual learning, incorporate long-range radios and solar power, and optionally capture signed, privacy-aware evidence snapshots. Overall, the results indicate that a small, rugged, and inexpensive edge node can meaningfully shorten detection-to-notification time and complement satellite and patrol methods for forest protection.

## Compliance with ethical standards

### *Disclosure of conflict of interest*

Authors do not have any conflict of Interest.

## References

- [1] Andreadis, A., Giambene, G., & Zambon, R. (2021). Monitoring illegal tree cutting through ultra-low-power smart IoT devices. *Sensors*, 21(22), 7593. <https://doi.org/10.3390/s21227593> MDPI
- [2] Avazov, K., Hyun, A. E., Sami S, A. A., Khaitov, A., Abdusalomov, A. B., & Cho, Y. I. (2023). Forest fire detection and notification method based on AI and IoT approaches. *Future Internet*, 15(2), 61. <https://doi.org/10.3390/fi15020061> MDPI
- [3] CIFOR-ICRAF. (2024, December 18). A year of hope and hardship for the world's forests. <https://forestsnews.cifor.org/90550/> (cites INTERPOL estimates that illegal logging accounts for 15–30% of global timber production). CIFOR-ICRAF Forests News

- [4] Chan, C. C., Alvi, S. A., Zhou, X., Durrani, S., Wilson, N., & Yebra, M. (2024). A survey on IoT ground sensing systems for early wildfire detection: Technologies, challenges and opportunities. arXiv:2312.10919. <https://arxiv.org/abs/2312.10919> arXiv
- [5] da Silva, D. Q., dos Santos, F. N., Filipe, V., Sousa, A. J., & Oliveira, P. M. (2022). Edge AI-based tree trunk detection for forestry monitoring robotics. *Robotics*, 11(6), 136. <https://doi.org/10.3390/robotics11060136> MDPI
- [6] Liu, Y., Cheng, Z., Liu, J., Yassin, B., Nan, Z., & Luo, J. (2020). An energy-efficient fall detection method based on FD-DNN for elderly people. *Sensors*, 20(16), 4796. (MPU6050 free-fall interrupt behavior discussed.) <https://doi.org/10.3390/s20164796> PMC
- [7] Mporas, I., Perikos, I., Kelefouras, V., & Paraskevas, M. (2020). Illegal logging detection based on acoustic surveillance of forest. *Applied Sciences*, 10(20), 7379. <https://doi.org/10.3390/app10207379> MDPI
- [8] Puttapirat, P., Woradit, K., Hesse, H., & Bhatia, D. (2024). FireFly project: UAV development for distributed sensing of forest fires. In *Proceedings of the 2024 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 594–601). IEEE. <https://doi.org/10.1109/ICUAS60882.2024.10556892> Enlighten Publications
- [9] Silva, R. L. S., Pessin, G., Kozakevicius, A. J., Rodrigues, R. C., & Prado, A. (2021). Wearable edge AI applications for ecological environments. *Sensors*, 21(15), 5041. <https://doi.org/10.3390/s21155041> PMC
- [10] Ahmad, S. F., & Singh, D. K. (2019). Automatic detection of tree cutting in forests using acoustic properties. *Journal of King Saud University-Computer and Information Sciences*, 31(1), 120–129. <https://doi.org/10.1016/j.jksuci.2019.01.016> ScienceDirect
- [11] Mou, A., & Milanova, M. (2024). Performance analysis of deep learning model-compression techniques for audio classification on edge devices. *Sci*, 6(2), 21. <https://doi.org/10.3390/sci6020021> MDPI
- [12] Avazov, K., et al. (2023). Forest fire detection and notification method based on AI and IoT approaches. *Future Internet*, 15(2), 61. <https://doi.org/10.3390/fi15020061> MDPI
- [13] Chandak, K., Sanadhya, A., Gohil, J., Trivedi, R., Parikh, P., Chauhan, M., Patel, K., & Prajapati, H. (2025). Electromyography operated soft finger-like actuator for prosthesis. *International Journal on Interactive Design and Manufacturing*, 19(3), 2283–2302. <https://doi.org/10.1007/s12008-024-01911-1>
- [14] Gadhavi, P., Shah, P., Trivedi, R., & Parikh, P. (2025). Analysis of Li Ion Battery using Computational Fluid Dynamics. 1–5. <https://doi.org/10.1109/amathe65477.2025.11081206>
- [15] Gohil, J. A., Trivedi, R. R., & Parikh, P. A. (2023). Development Of A Remotely Operated 3D Printed Robotic Hand Using Electromyography. *AIP Conference Proceedings*, 2946(1). <https://doi.org/10.1063/5.0178508>
- [16] Joshi, K. D., Maheshwari, N., Patel, H., & Parikh, P. A. (2025). Divyawear-A Wearable Haptic Cueing System for the Visually Impaired Indian People. *International Journal of Computer Applications*, 186(79), 975–8887. <https://doi.org/10.5120/ijca2025924707>
- [17] Kannan, R., Shah, P., & Parikh, P. (2025). DESIGN, SIMULATION, AND WALKING PATTERN ANALYSIS OF A PROSTHETIC LEG MADE FROM RECYCLED PET PLASTIC FOR SUSTAINABLE AND AFFORDABLE HEALTHCARE SOLUTIONS. *Indian Journal of Applied Research*, XV(03), 45–48. <https://doi.org/10.36106/ijar/0307952>
- [18] Parikh, P. A., Joshi, K. D., & Trivedi, R. (2023). Face Detection-Based Depth Estimation by 2D and 3D Cameras: A Comparison. *2022 28th International Conference on Mechatronics and Machine Vision in Practice, M2VIP 2022*, 1–4. <https://doi.org/10.1109/M2VIP55626.2022.10041072>
- [19] Parikh, P. A., Tolaney, R. S., Shah, C. M., Head, A. P., & Students, U. G. (2016). Wireless Multichannel Data Acquisition System using ARDUINO and MATLAB GUI. *International Journal of Engineering Development and Research*, 4, 1711–1716. [www.ijedr.org](http://www.ijedr.org)
- [20] Parikh, P. A., Trivedi, R., & Dave, J. (2020). Trajectory planning for the five degree of freedom feeding robot using septic and nonic functions. *International Journal of Mechanical Engineering and Robotics Research*, 9(7), 1043–1050. <https://doi.org/10.18178/ijmerr.9.7.1043-1050>
- [21] Parikh, P. A., Trivedi, R., & Joshi, K. D. (2020). Vision-Based Trajectory Planning for a Five Degree of Freedom Assistive Feeding Robotic Arm Using Linear Segments with Parabolic Blend and Cycloid Functions. In *Mechatronics and Machine Vision in Practice 4* (pp. 193–206). Springer International Publishing. [https://doi.org/10.1007/978-3-030-43703-9\\_16](https://doi.org/10.1007/978-3-030-43703-9_16)
- [22] Parikh, P. A., Trivedi, R., & Joshi, K. D. (2023). Optimising inverse kinematics algorithm for an indigenous vision-based feeding serial robot using particle swarm optimisation and hybrid genetic algorithm: a comparison

- Optimising inverse kinematics algorithm for an indigenous vision-based feeding serial robot. *Int. J. Advanced Mechatronic Systems*, 10(2), 88–101. <https://doi.org/https://doi.org/10.1504/IJAMECHS.2023.131332>
- [23] Parikh, P. A., Trivedi, R. R., & Joshi, K. D. (2021). Trajectory planning of a 5 DOF feeding serial manipulator using 6th order polynomial method. *Journal of Physics: Conference Series*, 1921(1). <https://doi.org/10.1088/1742-6596/1921/1/012088>
- [24] Parikh, P., Sharma, A., Trivedi, R., Roy, D., & Joshi, K. (2025). Performance evaluation of an indigenously-designed high performance dynamic feeding robotic structure using advanced additive manufacturing technology, machine learning and robot kinematics. *International Journal on Interactive Design and Manufacturing*, 19(2), 909–937. <https://doi.org/10.1007/s12008-023-01513-3>
- [25] Parikh, P., Sheth, S., Vasani, R., & Gohil, J. K. (2018). Implementing Fuzzy Logic Controller and PID Controller to a DC Encoder Motor - “a case of an Automated Guided Vehicle.” *Procedia Manufacturing*, 20, 219–226. <https://doi.org/10.1016/j.promfg.2018.02.032>
- [26] Parikh, P., Trivedi, R., Dave, J., Joshi, K., & Adhyaru, D. (2024). Design and Development of a Low-Cost Vision-Based 6 DoF Assistive Feeding Robot for the Aged and Specially-Abled People. *IETE Journal of Research*, 70(2), 1716–1744. <https://doi.org/10.1080/03772063.2023.2173665>
- [27] Parikh, P., Trivedi, R., & Joshi, K. (2023). Continuous trajectory planning of a 6 DoF feeding robotic arm using a novel multi-point LSPB algorithm. 2022 28th International Conference on Mechatronics and Machine Vision in Practice, M2VIP 2022, 1–6. <https://doi.org/10.1109/M2VIP55626.2022.10041082>
- [28] Parikh, P., Trivedi, R., Sharma, A., Dalwadi, N., & Joshi, K. (2025). Kinematics of an indigenous 3D printed 6 DoF feeding serial manipulator: Forward and inverse kinematics, singularity and workspace analysis. *AIP Conference Proceedings*, 3227(1). <https://doi.org/10.1063/5.0243297>
- [29] Parikh, P., Vasani, R., & Sheth, S. (2016). Velocity analysis of a DC brushed encoder motor using Ziegler-Nichols algorithm: A case of an automated guided vehicle. *Indian Journal of Science and Technology*, 9(38). <https://doi.org/10.17485/ijst/2016/v9i38/100884>
- [30] Parikh, P., Vasani, R., Sheth, S., & Gohil, J. (2017). Actuation of Electro-Pneumatic System using MATLAB Simulink and Arduino Controller-A case of a Mechatronics systems Lab. In Dr. Brijesh Iyer (Ed.), *Proceedings of the International Conference on Communication and Signal Processing 2016 (ICCASP 2016)* (Vol. 137, pp. 59–64). ATLANTIS Press, Springer Nature. <https://doi.org/10.2991/iccasp-16.2017.10>
- [31] Patel, A., Parikh, P., & Shah, P. (2025). DESIGN AND DEVELOPMENT OF A TANGIBLE AND DIGITAL LUNGS EXERCISER FOR THE ATHLETES. *Indian Journal of Applied Research*, XV(03). <https://doi.org/10.36106/ijar/7208051>
- [32] Patel, D. D., Patel, T. J., & Parikh, P. A. (2017). Modeling and Control of Inherently Unstable SIMO System. *International Journal of Innovative Research in Technology*, 3(8), 8–12. <https://doi.org/https://doi.org/10.5281/zenodo.16811424>
- [33] Sanadhya, A., Chandak, K., Gohil, J., Trivedi, R., Roy, D., Parikh, P., Chauhan, M., & Patel, K. (2025). Advancements in finger prosthesis: A design using shape memory alloy actuation mechanism. *Results in Engineering*, 27. <https://doi.org/10.1016/j.rineng.2025.105050>