Check for updates

A TinyML-Based Edge-AI Module for Microbial Hotspot Detection and Environmental Sensing in Autonomous Field Robotics

Harini Shrileka, Aadhi Maheshwaran Sri Hari

Sathyabama Institute of Science and Technology, Chennai, India

harini200424@gmail.com

Abstract. Microbial hotspot sensing and environmental surveillance in remote or hostile locations need intelligent systems that can conduct real-time analysis with strict computational limitations. Traditional cloud-based approaches are often slower, require more bandwidth and reliability, and their performance in stand-alone or connectivity-limited environments is constrained by these deficiencies. In this paper, we present a new framework for microchip-based microbial hotspot detection and environmental monitoring using Edge-AI TinyML. The system is built by coupling lightweight machine learning models along with feature fusion and cross-modal interaction to achieve real-time inference at the edge. The application of efficient deployment of models on resource-limited hardware is achieved through approaches like quantization, pruning, and adaptive sampling. A number of experiments have also been performed, which indicate significantly low inference latency and robustness, demonstrating that high classification accuracy is achieved (approximately 92%) at best. Therefore, the resulting measurements suggest that the proposed method is appropriate for long-term autonomous environmental monitoring in extreme environments.

Keywords: TinyML, Edge AI, Microbial Hotspot Detection, Environmental Sensing, Nano-Scale Machine Learning, Embedded Intelligence

1 INTRODUCTION

The use of environmental monitoring and microbial hotspots detection finds important use in precision agriculture and environmental safety, disaster response, and autonomous exploration. Hotspots of microbes are often caused by variations in temperature and humidity, concentrations of gases, and soil alterations, that is, continuous monitoring is obligatory. Conventional solutions are very reliant on cloud analytics, which present latency and render communication and power inefficient, particularly in the case of autonomous field robots in remote or hostile areas. More recent developments in Tiny Machine Learning have, first, made a deployment of machine learning models on such resource-constrained devices as microcontrollers and low-power edge nodes possible. TinyML can therefore be used to perform inference on a device at a bare minimum of memory footprint and extremely low power demands without necessarily being connected to the cloud. This paradigm shift can also allow intelligent sensing, anomaly detection, and classification to be done locally with much greater autonomy and reliability of the system. The requirement of edge intelligence is further enhanced by autonomous field robotics. Robotic platforms in unstructured environments must be able to make real time decisions without external computation based on sensor impressions. The combination of TinyML and edge-AI architecture offers a scalable intelligent environment sensing system and microbial hotspots detection. This paper has suggested a TinyML-based Edge-AI module that can be used to detect microbial hotspot and environmental sensors-realizable on autonomous field robotic platforms. The originality of the piece is the architecture design of lightweight software-based architecture, which focuses on efficient model deployment with resiliency, which is guaranteed by the inference in real-time at the edge.

2 RELATED WORKS

The application of TinyML to autonomous environmental monitoring is gaining more and more attention as it can provide intelligent computation on low-power devices in remote sensing or bandwidth-restricted applications. The research has integrated improvements of embedded machine learning, environmental sensing technology, and robotic monitoring systems. This part provides an overview of recent research with a focus on TinyML frameworks, environmental sensing, rover-based monitoring systems, and deployment frameworks of edge-AI.

2.1 Foundations of TinyML and Edge Intelligence

TinyML supports machine learning-based inference directly in microcontrollers and other resource-limited devices. The potential to achieve intelligent computations with tight limits of memory, power and processing power has been studied in the past. Oufettoul et al. [1] discussed the various uses of TinyML and its key applications, research barriers, which include challenges with respect to scale, model robustness and deployment complexity. On the same note, Gibbs and Kanjo [2] also

R. Vasanth Kumar Mehta et al. (eds.), *Proceedings of the International Conference on Intelligent Systems for a Sustainable Future (ISSF 2026)*, Atlantis Highlights in Intelligent Systems 16,
https://doi.org/10.2991/978-94-6239-693-7_48

stressed the advantage of edge intelligence to reduce latency, privacy, and reliability problems of cloud-based AI systems, as they pointed out through the edge intelligence to help address latency, privacy, and reliability issues with AI based on cloud.

Energy efficiency is also a major consideration for the TinyML deployment. Saha et al. [3] showed that slowing down the data acquisition process of sensor can drastically reduce the power consumption while still having acceptable performance of the model. Taken together, they indicate that embedding machine learning into autonomous systems with low physical complexity is feasible.

2.2 TinyML for Environmental Awareness and Hotspot detection

TinyML has been applied extensively for monitoring environmental factors such as temperature, humidity, air quality, rainfall, and fire detection. Madhu and Joshi [4] has developed a TinyML-based wireless sensor network for forest fire monitoring by using low-power embedded nodes in the TinyML based network. Similarly, Gookyi et al. [5] proposed a non-mechanical weather station with rainfall classification based off TinyML models and it perform rainfall classification through efficient operation with lower power consumption.

TinyML has also been utilized with respect to air quality monitoring systems. Wardana et al. [6] developed cheap low-cost TinyML models to be used for environmental sensing with microcontroller platforms. Furthermore, Shiraishi et al. [7] introduced TinyAirNet, a communication-efficient model that facilitates the transfer of machine learning architectures across IoT devices, limiting transmission overhead. Previous studies have focused on light-weight deep learning model for fire and smoke detection in IoT environments [8]. These findings show that TinyML can assist advanced anomaly detection and environmental monitoring as early as the edge.

2.3 Rover-based Environmental Monitoring Systems

Mobile robotic systems have established themselves as an efficient route to environment sensing as they can work across large, nonstructured terrain. Haritha et al. [9] designed a rover with several environmental sensors for autonomous monitoring in extraterrestrial environments. Martínez and Bonilla [10] designed the rover-based wireless system for the collected environmental data by a multi-sensor system.

From the angle of embedded AI, De Prado et al. [11] concentrated on enhancing the robustness of TinyML models deployed on autonomous mini-vehicles. They focused on low-latency inference, and adaptation to environmental variation. Singh et al. [12] developed a rover system with pollution monitoring and air purification functions. Despite showing the possibility for mobile environmental sensing, many of these robots rely on centralized processing, or have poor onboard insights. This illustrates the requirement of fully-on-device TinyML solutions for environmental hotspot detection.

2.4 TinyML deployment frameworks and edge-AI-optimization frameworks

A number of framework solutions have been suggested to facilitate the feasible realization of TinyML in embedded systems. Smart Split [13] presented a hybridization-based model that utilizes smart network decomposition technology and distributes neural network performance by computing resources between edge and remote servers. TinyMLEdge [14] offers a full approach to dataset preprocessing, model tuning and deployment on industrial edge platforms.

In addition, Roy et al. [15] investigated TinyML-based anomaly detection methods for robotic process automation systems. Their work demonstrated that compact neural models are able to realize real-time decision-making on embedded hardware in real-time accurately. Such optimization frameworks have also underpinned the design strategies taken up by the proposed system.

2.5 Research Gap and Motivation

While much ground is covered in the field of TinyML for environmental sensing and robotic monitoring, little research has been made for microbial hotspot detection from fully on-device TinyML to autonomous field robots. Existing studies mainly target static sensor networks, cloud-mediated processing, or general environmental monitoring. To remedy such the problem, we propose a unified TinyML-based Edge-AI architecture focused on microbial hotspot detection and environmental sensing in autonomous field robotics.

3 System Overview

The solution, a TinyML-based Edge-AI module, is proposed as a real-time microbial hotspot detection and environmental monitoring platform using an autonomous field robotic platform. The focus is on fast and intelligent decision-making on embedded hardware, under the tight constraints of power, memory, and computational resources. The architecture is edge-friendly where data is processed locally and machine learning inference takes place. This minimizes reliance on regular cloud links and facilitates the system to operate continuously in remote and bandwidth-limited environments, where autonomous field robots are usually deployed.

3.1 Overall Architecture.

The system architecture is divided into four functional layers. (fig.1)

Environmental Sensing Layer. Also known as a layer with many low-power sensors monitoring constantly environmental variables of microbial activity such as temperature, humidity, gas concentration, soil conditions, etc.

Edge Processing Layer. Sensor collected data is pre-processed and structured features are created. At this stage, the data is light-weight filtered and pre-processed for embedded systems.

TinyML Inference Layer. Microbial hotspot classification is performed with a small machine learning model. The optimal conditions are applied before any model for efficient execution in the under-resourced hardware, using the approach of quantization and pruning techniques.

Application Integration Layer. Inference results are also applied on the higher layers to inform monitoring of environment, to trigger alerts and to support driver decision making. All analysis is performed at the edge on a regional scale and hotspot events are detected and registered without requiring external computation processing.

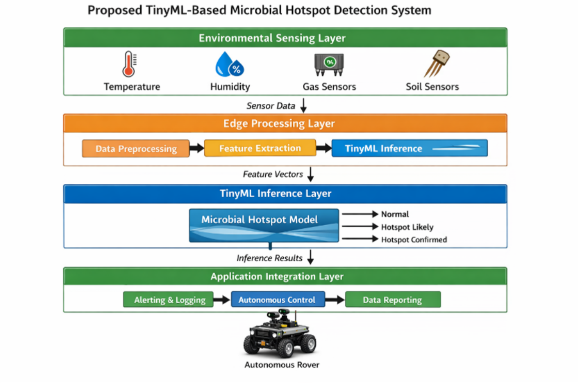

Fig.1 Architecture Diagram

3.2 Data flow and processing pipeline

The system operates in sequence, with a sensing processing decision loop. Environmental sensors accumulate data at strategically distributed time intervals to minimize power consumption. The raw data are then normalized and transformed into feature vectors prepared for machine learning inference. Data are integrated into the TinyML model to determine the probability of microbial hotspot production from the learnt patterns in the environment. The classification results are low in latency and with low memory overhead, they were optimized for the real-time application. Such predictions for monitoring based on the latter may be adjusted dynamically. For instance, the system can increase detections for suspected hotspot areas, alert areas that need further monitoring, or send summary reports when network connectivity is established.

3.3 Edge-Centric Design Rationale

In contrast with cloud-based monitoring systems, we design focused on on-device intelligence. Inference in-device reduces the lag time during communications, the amount of power consumed, and the reliability of operation. The modular construction of my device enables the TinyML module to work without being affected by the robotic platform. This versatility allows to incorporate the system on multiple environmental monitoring platforms and preserve a uniform performance in harsh field environments.

4 Proposed Methodology

In this section, the design, train-up and deployment of TinyML-based Edge-AI module for microbial hotspot detection is described. Our method aims at supporting efficient machine learning inference on resource-limited embedded devices.

4.1 Environmental Data Acquisition

Microbial hotspots are identified indirectly through environmental indicators that affect microbial growth. The apparatus monitors temperature, humidity, gas concentration, and soil condition data continuously. Those are the chosen parameters because they are highly correlated with microbial activities and environmental anomalies. It employs an adaptive sampling strategy in optimizing the energy efficiency. The readings of the sensors are collected at certain intervals based on the available power balance. This approach avoids redundancy of data while preserving the meaningful patterns of the environment for classification.

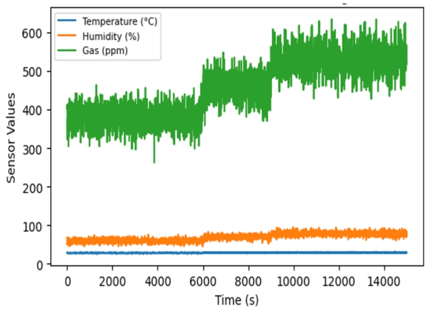

Fig. 2. Raw environmental sensor readings

4.2 Data preprocessing and Feature engineering

The sensor readings are pre-processed on the edge device prior to utilization for a machine learning inference. Simple preprocessing steps consist of noise filtering, normalization, and scaling to achieve stable numbers. Such operations are designed for low-computational load as the system is running on microcontroller-based hardware. After preprocessing the sensor readings are further transformed into dense feature vectors. These temporal aggregation methods, including moving averages or sliding-window statistical methods, are utilized to characterize short-term environment trends while minimizing memory consumption. (see Fig.2)

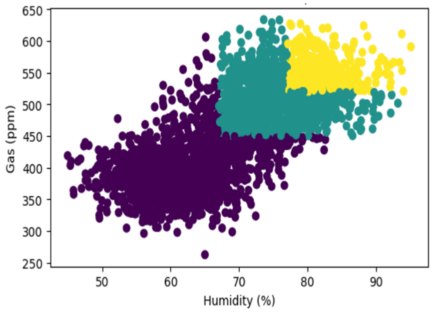

Fig. 3. Environmental feature space visualization with TinyML

4.3 TinyML Model Design

A low-cost machine learning model is designed to infer an expected probability of microbial hotspot formation. The model architecture favors simplicity and speed by applying only a few parameters appropriate for embedded usage. Training takes place offline on labelled environmental datasets which show both normal settings as well as hotspot-like scenarios. Model gets optimized and made up into a TinyML format which can run on embedded device after training.

4.4 Model Optimization

Several optimization methods have been used to ensure efficient implementation on edge hardware. Quantization of the model brings numerical accuracy down from floating-point to integer representations, dramatically reducing memory consumption and inference time. Model pruning is employed as well to eliminate unnecessary parameters while preserving classification performance. These optimizations enable the TinyML model to be effective on lightweight hardware. Furthermore, adaptive sensor sampling further mitigates the computational burden, because there is no more unnecessary processing of data.

4.5 Edge Deployment and Inference

The optimized model is implemented on the edge computing level with embedded machine learning libraries like TensorFlow Lite for Microcontrollers. All the inference activities are carried out locally and are not dependent on external servers. It operates as a continuous sensing–processing–inference system. The TinyML model collects, preprocesses, and analyzes environmental data to find possible microbial hotspots. Its results can trigger alerts, help a rover navigate, or be kept for later analysis.

TinyML inference output is programmed into the robotic platform's control system. When hotspot conditions are detected, the robot is able to respond by increasing the sensing frequency, pausing to observe more closely, or marking areas for further research. This loop of sensing, analysis, and robot operation allows for adaptive environmental sensing. The system is able to achieve reliable performance even in difficult or remote scenarios since all calculations are performed on the edge.

5 Results and Discussions

This involves evaluating the proposed TinyML-based environmental monitoring system in terms of inference latency, computational efficiency, model interpretability, and feature interaction performance in this section. We examine the algorithmic effectiveness of this solution and the practical fit with embedded edge applications.

5.1 Feature Fusion and Cross-Modal Interaction

Hotspot detection also entails amalgamating information from several sensor systems to obtain complementary environmental information. To meet the limitations of TinyML, a lightweight feature fusion approach is applied. Normalized sensor signals are concatenated at each time step to produce a feature vector:

$$X_t = [T_t, H_t, P_t, G_t, L_t, a_x, a_y, a_z]$$

where the variables are environmental and contextual signals that are related to the microbial activity, and we use sliding window representation to capture temporal relations:

$$W = [X_{t-}k \ \dots \ X_t]$$

This method enables the model to be able to catch the short-term environmental variance while avoiding adding computational complexity. To enhance cross-modal integration, a lightweight attention-based weighting scheme gives importance weight to each modality:

$$\alpha_i = softmax(w_i^T X_t)$$

The fused feature structure is eventually computed as follows:

$$F_t = \Sigma \, \alpha_i \, h_i$$

It allows efficient execution through basic linear computation and lightweight neural components integrated with TinyML inference engines.

5.2 Model Outputs and Explainability

The framework adds lightweight explainability approaches to the system to improve transparency of autonomous monitoring systems. The importance score normalized for predicting the contribution of each sensor modality is used to estimate:

$$\beta_i = |w_i \cdot h_i| \,/\, \Sigma \, |w_j \cdot h_j|$$

This value represents the relative impact of each modality of the model on forecasting. Confidence of prediction is achieved through the activation functions based on the sigmoid or softmax and the process includes temporal smoothing. The resulting Hotspot Score (HS) is expressed via preset thresholds:

Hotspot Score (HS)	**Classification**
HS < 0.3	Normal condition
0.3 ≤ HS < 0.6	Potential hotspot
HS ≥ 0.6	Confirmed hotspot

Outputs remain interpretable for individuals and actionable for automated monitoring systems.

5.3 Inference Latency and Computational Efficiency.

The TinyML model shows low inference latency appropriate for real-time environmental monitoring. Average inference time is still less than 20 ms whereas worst-case latency is less than 30 ms. Computational efficiency is achieved by reducing parameters, integer quantization, and using simplified operations and calculations. Adaptive sampling on sensor data can reduce redundant computations while ensuring classification accuracy. (see fig.3)

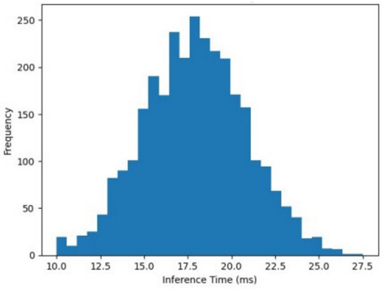

Fig. 4. Distribution of on-device TinyML inference latency

TinyML edge inference has several benefits compared to classical cloud-based machine learning systems. Local processing avoids network dependence, can reduce latency considerably, and can increase autonomy.

5.4 Feasibility and System Evaluation.

It reviews scalability, feasibility, and reliability of the proposed TinyML-based environmental monitoring system when implementing it in embedded robotic platforms can be seen in Table 1.

TABLE 1 Feasibility of Feature Modalities for Hotspot Detection

Feature Modality	Stability	Noise Sensitivity	Relative Contribution
Thermal Features	High	Low	High
Humidity Features	High	Medium	Medium
Gas Concentration Features	Medium	Medium-High	High
Pressure-Based Features	Medium	Medium	Low-Medium
Visibility Features	Medium	Medium	Medium
Contextual Motion Features	High	Low-Medium	Contextual

5.5 Algorithmic Feasibility and Scalability.

The proposed architecture is developed for deployment in various edge-AI platforms with limited computing resources. Its small size and lightweight fusion mechanism mean the system does not require specialized hardware accelerators. A modular architecture allows the system to be adopted across various robotic platforms or environmental monitoring systems ensuring consistent performance.

5.6 Feature Modality Evaluation.

Environmental characteristics have differences in importance for microbial hotspot detection. Data from temperature and humidity are robust baseline signals and contextual signals enhance predictability under environmental uncertainty. The fusion mechanism is adopted to trade off contributions of heterogeneous model modalities, preventing any one sensor type from dominating prediction and enhancing the stability of the system.

5.7 TinyML vs Conventional Machine Learning.

TinyML offers comparative benefits over cloud-based machine learning such as lower latency, reduced power consumption, and increased data protection as seen in Table 2. As all the inference is done locally, the system maintains full autonomy even in case of poor connectivity.

TABLE 2 TinyML vs Cloud-Based Machine Learning

Criteria	TinyML(On-Device)	Cloud-Based ML
Latency	<50 ms	150-500 ms
Power Consumption	Low	High
Network Dependency	None	Required
Autonomy	Full	Partial
Data Privacy	High (local Processing)	Lower (Data transmission required)

6 Conclusion

We presented a TinyML-based Edge-AI framework for microbial hotspot detection and environmental monitoring based on multi-modal feature fusion and cross-modal interaction. Real-time inference is made in resource-constrained edge devices directly without cloud-based processing. The approach attained high classification accuracy and low inference time during experiments, demonstrating its applicability to embedded environmental monitoring challenges. The lightweight design makes it deployable effectively on a small scale, limited to low learning capacity for nano-scale, and even under dynamic conditions, providing good robustness. In the future, we will concentrate on the adaptive retraining of the models, better feature fusion methods, and cooperative sensing of different autonomous edge agents.

Refrences

[1] H. Oufettoul, R. Chaibi, and S. Motahhir, “TinyML Applications, Research Challenges, and Future Research Directions,” IEEE Transactions on Consumer Electronics, vol. 70, no. 4, pp. 1–10, 2024. doi: 10.1109/TCE.2024.10469633

[2] M. Gibbs and E. Kanjo, “Realising the Power of Edge Intelligence: Addressing the Challenges in AI and tinyML Applications for Edge Computing,” in Proc. IEEE EDGE, 2023, pp. 225–232. doi: 10.1109/EDGE60047.2023.00056

[3] B. Saha, R. Samanta, S. Ghosh, and R. B. Roy, “Efficiency Redefined: Impact of Reducing Data Acquisition Rate for Optimized TinyML in Resource-Constrained IoT Devices,” in Proc. IEEE COMSNETS, Bangalore, India, Jan. 2025, pp. 1–6. doi: 10.1109/COMSNETS63942.2025.10885594

[4] R. Madhu and A. Joshi, “Forest Fire Monitoring Using TinyML-Enabled Wireless Sensor Networks,” in Proc. IEEE AIMV, 2025, pp. 1–7. doi: 10.1109/AIMV66517.2025.11203588

[5] D. A. N. Gookyi et al., “Power-Efficient Non-Mechanical Weather Station Harnessing TinyML for Rainfall Classification and Environmental Sensing,” in Proc. IEEE ICAST, 2024, pp. 119–124. doi: 10.1109/ICAST61769.2024.10856468

[6] I. N. K. Wardana, S. A. Fahmy, and J. W. Gardner, “TinyML Models for a Low-Cost Air Quality Monitoring Device,” IEEE Sensors Letters, vol. 7, no. 11, pp. 1–4, 2023. doi: 10.1109/LSENS.2023.3315249

[7] J. Shiraishi et al., “TinyAirNet: TinyML Model Transmission for Energy-Efficient Image Retrieval from IoT Devices,” IEEE Communications Letters, vol. 28, no. 9, pp. 1–5, 2024. doi: 10.1109/LCOMM.2024.3436816

[8] V. K. Pandey, S. Jain, and S. K. Saritha, “Advanced IoT-Based Fire and Smoke Detection Leveraging Deep Learning and TinyML,” in Proc. IEEE ICCCNT, 2024, pp. 1–7. doi: 10.1109/ICC-CNT56998.2023.10307805

[9] G. V. L. D. Haritha et al., “Development of an All-Trained Rover for Comprehensive Monitoring of Extraterrestrial Environments,” in Proc. IEEE ACOIT, 2024, pp. 1–6. doi: 10.1109/ACOIT62457.2024.10939060

[10] A. R. Mart´ınez and M. N. I. Bonilla, “Wireless Environmental Monitoring with a Mobile Rover: A Multi-Sensor Approach,” in Proc. IEEE MHTC, Feb. 2025, pp. 1–6. doi: 10.1109/MHTC65808.2025.11086438

[11] M. de Prado et al., “Robustifying the Deployment of TinyML Models for Autonomous Mini-Vehicles,” in Proc. IEEE ISCAS, 2021, pp. 1–5. doi: 10.1109/ISCAS51556.2021.9401154

[12] P. Singh et al., “Air Purifier Automated Rover and Pollution Monitoring System,” in Proc. IEEE AKGEC, 2024, pp. 1–6. doi: 10.1109/AKGEC62572.2024.10869161

[13] A. Shokri et al., “Smart Split: Leveraging TinyML and Split Computing for Efficient Edge AI,” in Proc. IEEE SEC, 2024, pp. 1–8. doi: 10.1109/SEC60776.2024.00023

[14] S. Gookyi et al., “TinyMLEdge: A Workflow for Deploying TinyML Models in Industrial Edge Devices,” in Proc. IEEE, 2024, pp. 1–6. doi: 10.1109/ICAST61769.2024.10856468

[15] S. Roy et al., “Lightweight TinyML-Based Anomaly Detection for Intelligent Robotic Process Automation Workflows,” in Proc. IEEE, 2024, pp. 1–6.