TinyML-driven Sensor Nodes for Energy-efficient Acoustic Event Detection in Pervasive Acoustic WSNs

Bibek B. Roy[1], Sushovan Das[2], and Uttam Kr. Mondal[1]

[1] *Vidyasagar University, Midnapore, WB, India,*
[2] *College of Engineering & Management, Kolaghat, WB, India*

https://doi.org/10.26636/jtit.2025.2.2084

Abstract — The process of sensing and transmitting acoustic signals by pervasive acoustic wireless sensor networks (PAWSNs) poses considerable energy challenges. These problems may be mitigated by filtering only relevant acoustic events from the sensor network. By reducing the number of acoustic events, the frequency of communication may be decreased, thereby enhancing energy efficiency. Although traditional machine learning models are capable of predicting relevant acoustic events by being trained on suitable data sets, they are impractical for direct implementation on resource-limited acoustic sensor nodes. To address this issue, this research introduces TinyML-based acoustic event detection (AED) models which facilitate efficient real-time processing on microcontrollers with scarce hardware resources. The study develops several TinyML models using an environmental dataset and evaluates their accuracy. These models are then deployed in hardware to assess their performance in terms of AED. Thanks to such an approach, only predicted events that exceed a certain threshold are transmitted to the base station via router nodes, which reduces the transmission burden, thus improving energy efficiency of PAWSNs. Real-time experiments confirm that the proposed method significantly improves energy efficiency and boosts node lifetime.

Keywords — acoustic event detection, energy efficiency, pervasive acoustic WSN, TinyML

1. Introduction

Pervasive acoustic wireless sensor networks (PAWSNs) [1], [2] are composed of ubiquitous sensors dedicated to monitoring various environmental settings. Such networks are essential in real-time applications, such as underwater monitoring, detection of anomalies in industrial devices, smart city infrastructure, and observation of wildlife. However, spatially distributed battery-operated acoustic sensors continuously capture, process and transmit acoustic signals, raising concerns about excessive energy consumption and reducing the overall lifespan of the network. The conventional approach relying on central processing of the sensed data is inefficient for large volumes of data produced by PAWSNs. Typically, each detected acoustic event is sent to a base station (BS) through router nodes using traditional routing algorithms. This results in significant amounts of energy being consumed to conduct the transmissions and causes network congestion. It also needs to be borne in mind that PAWSNs, which operate in various environments, are faced with distinctive acoustic conditions.

Although traditional ML models could address this diversity by predicting or filtering acoustic events to lower transmission costs, they require large amounts of computing resources which are often unavailable in battery-powered leaf nodes that offer limited processing power. Therefore, this study introduces a lightweight, decentralized method that reduces data transmission while preserving high AED prediction accuracy.

This work proposes a TinyML-powered AED model that enables lightweight ML models to run on battery-operated microcontrollers with limited hardware, as a low-power real-time solution. Due to this, such devices as the Raspberry Pi platform are capable of predicting specific acoustic events at leaf nodes and only send, to the base station, the detected filtered events, thus significantly lowering energy use by reducing the amount of data exchanged between leaf and router nodes. This decentralized method addresses the challenge posed by the pervasive acoustic environment, as it decreases transmission loads, extends the useful life of the network, and improves the overall effectiveness of the system.

This work also employs a context-aware TinyML model selection strategy, in which models are deployed and tested for optimal performance on leaf nodes, based on environmental conditions. By selecting models with the highest level of accuracy for AED, each leaf node effectively detects relevant acoustic events in its specific acoustic zone. Moreover, due to security concerns, such as intruders mimicking environmental sounds, TinyML models must be updated regularly based on newly collected data, reaching beyond the initial dataset. The base station periodically updates the models and resends them to specific leaf nodes to maintain long-term event detection accuracy.

The primary advantages of the proposed work include incorporating TinyML-driven AED models into leaf nodes, optimization of PAWSN efficiency by minimizing the transmission of unnecessary acoustic data and decreasing storage needs, all while preserving high accuracy and ensuring secure environmental monitoring. This study helps improve AI-driven lightweight TinyML models, with the aim of en-

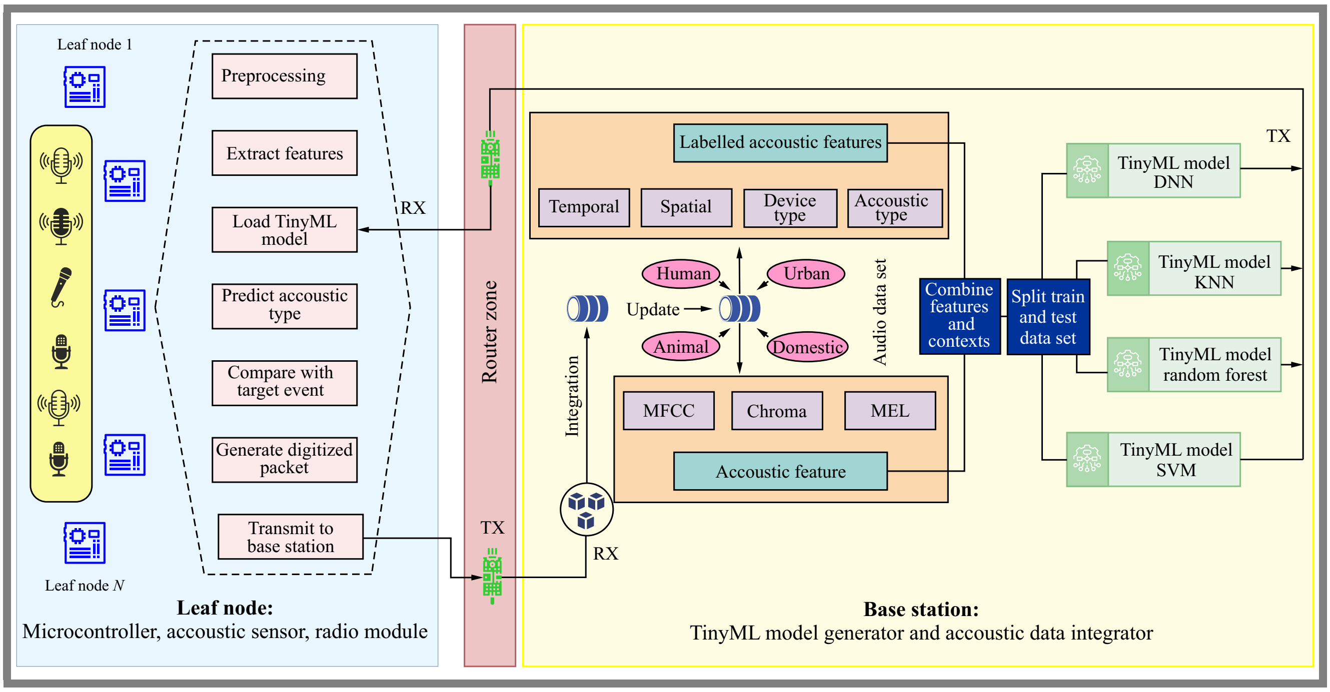

Fig. 1. PAWSN system architecture.

suring energy-efficient real-time detection of acoustic events with the use of resource-limited battery-powered acoustic sensors within PAWSN settings. The framework facilitates the prolonged deployment of PAWSNs across diverse monitoring applications by enabling nodes to dynamically adjust and update their models in response to changing acoustic environments, while simultaneously minimizing energy consumption.

2. Literature Review

Incorporating TinyML-based models into PAWSNs is crucial, as it allows real-time on-device processing in battery-operated, resource-limited acoustic sensor nodes [3]–[10]. Since the process of monitoring environmental conditions in real time requires acoustic data to be transmitted on a continuous basis, an activity that results in significant energy consumption [11]–[13], viable solutions must incorporate mechanisms to filter irrelevant data. Traditional ML models, which require considerable computational capabilities, are incompatible with resource-limited sensor nodes. In contrast, TinyML is designed to operate efficiently on such sensor nodes, using minimal computational resources and energy, especially to detect acoustic events with a high accuracy rate at the sensor node level [11].

Optimizing and selecting models, such as quantifying and deploying neural networks, is crucial for sensor nodes with limited resources. These models aim to balance computational efficiency with improved precision [13]. The challenges posed by the dynamic nature of acoustic environments require periodic updating of ML models to ensure long-term viability. Various learning methodologies are used to update TinyML models at the base station, improving PAWSN's ability to adapt effectively to changing environmental conditions [14]. Furthermore, the choice of low-power hardware is crucial for implementation in real-time environmental or industrial settings, such as detecting accidents involving workers through audio classification [15].

Although numerous studies [11], [13]–[15] focus on resolving issues such as achieving high accuracy and energy efficiency, many problems caused by challenging environmental conditions and noise still exist in practical applications. To address these challenges, future research should emphasize the development of TinyML model architectures by investigating supervised and unsupervised learning approaches.

Multiple studies have explored energy efficiency in PAWSNs by optimizing data collection, improving routing protocols, and incorporating ML-based techniques for data integration in BS. In [16], the authors introduced an energy efficient data aggregation algorithm that effectively collected data for unmanned aerial vehicles (UAVs). The authors of [17] presented an energy efficient data collection approach using autonomous underwater vehicles (AUVs) for underwater acoustic sensor networks, highlighting the progress in optimizing data transmission strategies.

In [18], the authors introduced a single-relay selective method for WSNs aimed at reducing energy use while maintaining reliable communication. Similarly, [19] explored energy efficiency in the industrial monitoring of WSNs with limited energy, highlighting the advantages of a cooperative communication protocol. The strategies implemented by these researchers promote more sustainable and durable sensor networks by reducing transmission overhead and optimizing resource use.

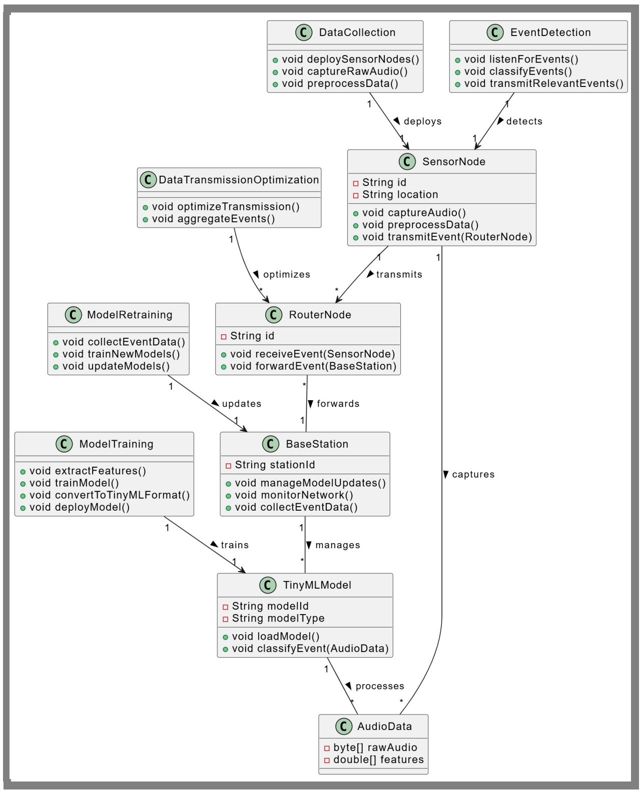

Fig. 2. PAWSN workflow with TinyML-driven sensor.

Incorporation of TinyML-based AED models into sensor nodes is becoming a trend for energy-efficient PAWSNs. Study [11] explored the use of RNN-based ML models in low-power sensor devices for AED, achieving real-time processing. In [14], the researchers presented a versatile and unsupervised TinyML framework designed to identify anomalies in industrial settings, incorporating on-device learning and training capabilities.

Although technological advancements have been introduced to sensors, communications, data processing, integration, and ML-based prediction, PAWSNs still face challenges related to energy efficiency and security surveillance during continuous environmental monitoring. Furthermore, complex processing of acoustic signals and handling of large amounts of acoustic data can negatively impact the accuracy of event detection processes.

3. Proposed Technique

Figure 1 illustrates the proposed TinyML-based PAWSN architecture which consists of three primary components: leaf nodes, a router zone, and a base station. The leaf nodes, located at the sensing edge, contain microcontrollers integrated with acoustic sensors and a radio module. Each leaf node performs local acoustic processing, including feature extraction (e.g. MFCC, chroma, MEL), TinyML model inference, and acoustic-type prediction. If the prediction aligns with a predefined target event, the node generates a digitized packet and transmits it to the base station through the router zone. These nodes are battery operated, and due to the use of compact TinyML models (with a code size of 200 KB, their power consumption remains low at 0.25 ... 0.45 W), allowing extended operational lifetimes depending on the duty cycle.

The router zone enables multihop or range-extended wireless communication using such platforms as ESP-Now or XBee. The BS acts as the central intelligence hub, typically a high-end GPU-based computer, as it integrates and labels the acoustic dataset, extracts contextual features (temporal, spatial, device type), splits the data for training and testing, and generates optimized TinyML models.

Although the PC is essential during model development and training, it can also be employed in field deployments for real-time integration and reconfiguration, although it can be replaced with lightweight embedded computing platforms in resource-constrained environments.

TinyML model development [20], [21] frequently favors the use of SVM, KNN, random forest, and dense neural networks (DNN) due to their empirical performance in acoustic classification tasks and their compatibility with microcontroller-class hardware constraints. These models are effective in classifying structured, low-dimensional acoustic features such as MFCCs, chroma, and MEL spectrograms, which are essential for identifying diverse environmental sound categories, as found in the ESC-50 dataset. Each algorithm was configured through preliminary tuning to balance inference accuracy and computational efficiency on resource-constrained hardware.

Specifically, we used an RBF kernel for SVM, KNN with $k = 5$, a random forest with 40 trees, and a DNN with two hidden layers (64 and 32 neurons, respectively). Classical models, such as decision trees and naive Bayes classifiers, although computationally lightweight, underperformed in generalization during initial evaluations. Random forest was selected over single decision trees for its ensemble robustness, while naive Bayes was excluded due to its strong independence assumptions, which are not well suited to the correlated nature of time-frequency audio features. The selected models strike an optimal trade-off between expressiveness and resource use, making them suitable for quantization and deployment on devices like the Seed XIAO ESP32S3 for energy-efficient, real-time inference in edge-based acoustic wireless sensor networks.

The data analysis begins by categorically separating various acoustic classes from 40 different environmental sounds, such as animal, urban, and human noises, using the ESC-50 dataset, complete with the appropriate labeling and target naming. Then, several TinyML models are trained using different classification algorithms such as K-nearest neighbors (KNN), support vector machine (SVM), random forest (RF), and DNN to handle the specific acoustic data present in the dataset. These models employ Mel-frequency cepstral coefficients (MFCCs) and chroma features to extract information from each category of environmental data. Next, the different TinyML models tailored for various environments are deployed to detect specific events. Upon sensing an acoustic event, the model forecasts its type. If the prediction score exceeds a predetermined threshold, only then is the detected

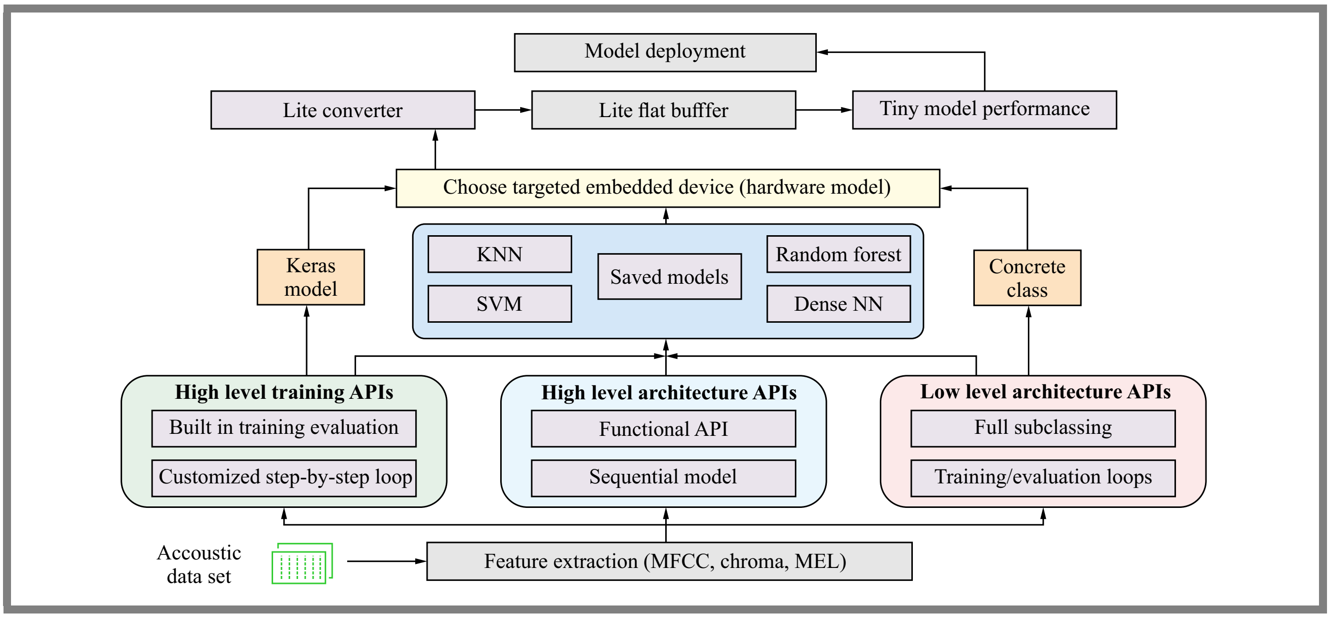

Fig. 3. Generation and deployment of the TinyML model for acoustic sensor nodes.

acoustic signal forwarded to router nodes for transmission to the BS.

Figure 2 illustrates the operational flow of the processes associated with the proposed system. Sensor nodes capture and preprocess raw audio data before transmitting filtered events to router nodes. The event detection module ensures accurate sound classification, while the data transmission optimization module reduces redundancy by aggregating and filtering events. The router nodes forward the filtered events to the BS which manages network operations, collects data for model updates, and optimizes the system's performance. The TinyML model is responsible for audio classification, with additional modules handling feature extraction, model training, and retraining to improve accuracy and adaptability over time.

The process of creating a TinyML model, as depicted in Fig. 3, begins with an acoustic dataset, where the raw audio data undergoes feature extraction using techniques such as MFCC, chroma, and MEL spectrograms. These extracted features serve as numerical representations of the sound data, which are then processed to train machine learning (ML) models.

The architecture leverages TensorFlow and Keras, providing multiple abstraction levels for model development. In high-level architectures, models can be structured using either the functional API or sequential model, offering flexibility in defining deep learning networks. For more advanced customization, the low-level API allows for full subclassing, where users can define custom layers and training loops. Training is facilitated through high-level training APIs, which include built-in evaluation methods and customized step-by-step loops for model optimization.

Once models are trained, they are saved for evaluation, where multiple algorithms such as SVM, KNN, RF and DNN are tested for performance. After selecting the best-performing model, the next step is to prepare it for embedded deployment. The model is optimized by choosing a targeted embedded device, with the next step consisting in converting it into a TensorFlow Lite (TFLite) model using the TFLite converter. To ensure that the model remains lightweight and efficient, the TFLite Flat buffer is used, facilitating model compression and quantization for tiny model's performance.

The final model is then deployed on edge devices, enabling low-power, real-time inference for applications such as IoT, smart devices, and real-time acoustic analysis. This process ensures that TinyML models are optimized to run on microcontrollers and embedded hardware while maintaining accuracy and efficiency.

3.1. TinyML-based Acoustic Event Detection

Algorithm 1 defines TinyML-based acoustic event detection with network simulation for efficient real-time sound classification and data transmission. Audio signals are processed in a loop over a predefined sensing period, classifying each captured sound using a TinyML model adapted to the sensor's zone. A confidence score is computed, and only high-confidence classifications are transmitted via XBee to the router nodes, which forward the data to the BS.

4. Mathematical Model

To quantify the energy efficiency of the proposed TinyML-driven acoustic event detection framework, this work develops a mathematical model considering event sensing, data transmission, and model retraining. Energy consumption is estimated using power specifications and operating durations derived from the Grove sensor, the XBee module, and the XIAO ESP32S3 microcontroller. Total energy is calculated as a weighted sum of the energy used during the sensing, pro-

Algorithm 1 TinyML-based AED for PAWSN.

Input: Acoustic signals from pervasive sensor zone
Output: Transmission of detected signals to BS
Parameters: T_{total} = total sensing duration, $T_{interval}$ = sensing interval, $S_{threshold}$ = prediction score threshold
procedure NetworkSimulation-TinyML( )
  Deploy acoustic sensor nodes in pervasive zones
  Establish wireless communication using XBEE
  Sense acoustic signals and preprocess (noise reduction, normalization, feature extraction)
  Train ML models using suitable dataset with various ML algorithms at BS
  Convert trained models to TinyML-compatible formats (TensorFlow Lite or pickle)
  Deploy models onto leaf nodes
  $t = 0$
  while $t < T_{total}$ **do**
    Sense acoustic signal from sensor zone
    Preprocess and extract MFCC features
    for each TinyML model **do**
      Perform AED using respective model
      Compute confidence score S
      if $S < S_{threshold}$ **then**
        Discard acoustic signal
      else
        Transmit detected event to routers
        Router node forwards data to the BS
      end if
    end for
    Wait for $T_{interval}$ before next sensing cycle
    $t = t + T_{interval}$
  end while
  BS collects, stores and integrates acoustic data
  Retrain TinyML models periodically
  Redeploy updated models to sensor nodes
end procedure

cessing, and communication phases, based on realistic task durations.

The total energy consumption of the PAWSN can be expressed as:

$$E_{total} = E_{leaf} + E_{router} , \tag{1}$$

where E_{leaf} is the total energy consumed by all leaf nodes and E_{router} stands for the total energy consumed by all router nodes.

4.1. Energy Model for Leaf Nodes

The energy consumed by a single leaf node is given by:

$$E_{leaf} = \sum_{i=1}^{T_{obs}/T_{event}} \left(P_{sense}\, T_{sense,i} + P_{process}\, T_{process,i}\right) + \sum_{i=1}^{T_{obs}/T_{event}} \left(P_{tx}\, T_{tx,i}\right) , \tag{2}$$

where: P_{sense}, $P_{process}$, and P_{tx} are the power consumption rates for sensing, processing, and transmission. $T_{sense,i}$, $T_{process,i}$, $T_{tx,i}$ are the corresponding time durations at interval i.

Since TinyML reduces transmission data volume, the effective energy consumption for a single-leaf node with TinyML can be expressed as:

$$E_{leaf}^{TinyML} = \sum_{i=1}^{T_{obs}/T_{event}} \left(P_{sense}\, T_{sense,i} + P_{process}\, T_{process,i}\right) + \sum_{i=1}^{T_{obs}/T_{event}} \left(P_{tx}\, T_{tx,i}\, \pi_{detect,i}\right) , \tag{3}$$

where $\pi_{detect,i}$ accounts for the probability of a transmission-triggering event being detected.

The probability of detecting an acoustic event in time interval i is as follows:

$$\pi_{\text{detect},i} = \frac{SNR_i}{SNR_i + \theta} \cdot \Phi(F_i, M) , \tag{4}$$

where SNR_i is the signal-to-noise ratio in time interval i, θ is the detection threshold, a system-defined parameter that determines the sensitivity to acoustic events, and $\Phi(F_i, M)$ is the TinyML model classification confidence function, which depends on:

- F_i – feature vector extracted from the acoustic signal at interval i,
- M – TinyML model used for event classification.

4.2. Energy Model for Router Nodes

The router nodes are responsible for receiving and forwarding data to the BS. The energy consumption per router node is given by:

$$E_{router} = \sum_{i=1}^{T_{obs}/T_{event}} \left(P_{rx}\, T_{rx,i} + P_{tx}\, T_{tx,i}\right) , \tag{5}$$

where P_{rx}, P_{tx} are the power consumption rates for receiving and transmitting data, while $T_{rx,i}$, $T_{tx,i}$ are the corresponding time durations at interval i.

With TinyML-based event detection, fewer events are transmitted from leaf nodes, reducing the forwarding burden on router nodes represented by:

$$E_{router}^{TinyML} = \sum_{i=1}^{T_{obs}/T_{event}} \left(P_{rx}\, T_{rx,i} P_{detect,i} + P_{tx}\, T_{tx,i} P_{detect,i}\right) , \tag{6}$$

4.3. Total Energy Consumption and Efficiency

The total energy consumption of the network is represented by:

$$E_{total} = \sum_{n \in N_{leaf}} E_{leaf,n} + \sum_{m \in N_{router}} E_{router,m} , \tag{7}$$

where N_{leaf} and N_{router} are the total numbers of leaf and router nodes, respectively.

With the proposed TinyML-based approach, the total energy consumption is:

$$E_{total}^{TinyML} = \sum_{n \in N_{leaf}} E_{leaf,n}^{TinyML} + \sum_{m \in N_{router}} E_{router,m}^{TinyML} , \quad (8)$$

Energy efficiency improvement can be quantified as:

$$\eta = \frac{E_{total} - E_{total}^{TinyML}}{E_{total}} \times 100\% , \quad (9)$$

where higher values of η indicate greater energy savings using the proposed model.

5. Experimental Results

To assess the effectiveness of the proposed system, this work sets up an acoustic wireless sensor network consisting of four sensor nodes, each equipped with a different training model for detecting various types of acoustic events. The network also includes four router nodes and a central BS. This study evaluates and compares different TinyML models with standard ML models in terms of hardware requirements and energy efficiency.

To establish a wireless acoustic sensor network, we use four Seeed XIAO ESP32S3 microcontrollers, each integrated with Grove sound sensors (LM358) and XBee radio modules as battery-operated leaf nodes. Four additional XBee modules serve as router nodes to enable multi-hop communication. A high-performance BS PC computer handles model training and evaluation.

The complete software stack is implemented in Python 3.10, using TensorFlow, Keras, scikit-learn, and Librosa for preprocessing, feature extraction, and model inference. Feature extraction (e.g., MFCCs) and model training are performed entirely on the BS, not on the sensor nodes. On the other hand, feature extraction for sensed audio signals is performed within the leaf nodes. The work deploys the trained TinyML models, quantized and memory-optimized (e.g., tflite, pkl) to the microcontrollers, which execute MicroPython-based scripts to perform real-time inference on acoustic inputs. Each node is configured to detect a specific class of sound events and transmits data only when a detection exceeds a confidence threshold, thus minimizing power consumption and communication overhead.

Table 1 summarizes hardware-related requirements for running TinyML and standard ML models on the proposed network.

5.1. System Output

Figures 4–5 describe the output for a single listening of the PAWSN for four different pervasive zones with different types of acoustic events at a specific time instant using TinyML models. The TinyML model is based on the RF algorithm, chosen for its performance, as shown in Figure 8.

Figure 4 illustrates that only high-confidence acoustic events are transmitted, reducing unnecessary transmissions and improving energy efficiency, although some lower-confidence

```
Sensor_Node_1 is capturing and processing acoustic event from pervasive Zone...1
Sensor_Node_1 detected 'dog' with confidence 0.71, forwarding to Router_2
Router_2 received 'dog' with confidence 0.71, forwarding to Base_Station
Base_Station received final event: 'dog' with confidence 0.71, logging data.
--------------------------------------------------------------------------------
Sensor_Node_2 is capturing and processing acoustic event from pervasive Zone...2
Sensor_Node_2 discarded low-confidence detection (0.64)
Sensor_Node_3 is capturing and processing acoustic event from pervasive Zone...3
Sensor_Node_3 detected 'clapping' with confidence 0.81, forwarding to Router_1
Router_l received 'clapping' with confidence 0.81, forwarding to Base_Station
Base_Station received final event: 'clapping' with confidence 0.81, logging data.
--------------------------------------------------------------------------------
Sensor_Node_4 is capturing and processing acoustic event from pervasive Zone...4
Sensor_Node_4 discarded low-confidence detection (0.70)
```

Fig. 4. Sensor node output during filtering of acoustic events according to the accuracy level (> 0.75).

```
Sensor_Node_l is capturing and processing acoustic event from pervasive Zone...1
Sensor_Node_l detected 'dog' with confidence 0.71 (All Events Mode), forwarding to Router_4
Router_4 received 'dog' with confidence 0.71, forwarding to Base_Station
Base_Station received final event: 'dog' with confidence 0.71, logging data.
--------------------------------------------------------------------------------------------
Sensor_Node_2 is capturing and processing acoustic event from pervasive Zone...1
Sensor_Node_2 detected 'frog' with confidence 0.64 (All Events Mode), forwarding to Router_4
Router_4 received 'frog' with confidence 0.64, forwarding to Base_Station
Base_Station received final event: 'frog' with confidence 0.64, logging data.
--------------------------------------------------------------------------------------------
Sensor_Node_3 is capturing and processing acoustic event from pervasive Zone...1
Sensor_Node_3 detected 'clapping' with confidence 0.81 (All Events Mode), forwarding to
Router_4 Router_4 received 'clapping' with confidence 0.81, forwarding to Base_Station
Base_Station received final event: 'clapping' with confidence 0.81, logging data.
--------------------------------------------------------------------------------------------
Sensor_Node 4 is capturing and processing acoustic event from pervasive Zone...1
Sensor_Node 4 detected 'rain' with confidence 0.70 (All Events Mode), forwarding to Router_l
Router_1 received 'rain' with confidence 0.70, forwarding to Base_Station
Base_Station received final event: 'rain' with confidence 0.70, logging data.
```

Fig. 5. Sensor node output during acoustic event processing without filtering.

```
Energy Consumption@Seeed XIAO ESP32S3 Microcontroller:
------------------------------------------------------
| Model Type          | Total Energy Consumption (J) |
------------------------------------------------------
| Proposed Model      | 3.08                         |
| All Events Detected | 5.19                         |
------------------------------------------------------
Energy Efficiency Improvement: 40.56%
```

Fig. 6. Energy consumption with and without filtering at a leaf node.

detections are discarded. In contrast, Fig. 5 shows that all detected events, regardless of the confidence level, are transmitted to the BS, ensuring a higher number of detections but potentially leading to incorrect detections due to the pervasive nature of the environment. This results in security threats and increased power consumption.

Figure 6 shows how the proposed model optimizes energy usage by reducing transmission overhead. The comparison of energy consumption shown in Fig. 6 further validates the effectiveness of the proposed model in optimizing power usage within a PAWSN framework. The energy consumption is calculated on the basis of the parameter values provided in Tab. 2. The figure illustrates that, for this particular instance of listening to a single event, the proposed model consumes only 3.08 J of energy, while a model without TinyML-based or context-aware filtering expends 5.19 J. This specific instance of PAWSN operation demonstrates a significant energy efficiency improvement of 40.56%, confirming that filtering low-confidence events effectively reduces unnecessary processing and transmission costs.

These results highlight the trade-off between comprehensive event detection and energy conservation, reinforcing the advantages of the proposed model for resource-constrained WSNs.

Tab. 1. Minimum hardware requirements for running TinyML and standard ML models.

Model	CPU	Memory	Power	Seeed XIAO ESP32S3 specification
TinyML models (~200 KB)				
Tiny-DNN	240 MHz	300 – 500 KB	0.25 – 0.35 W	240 MHz, 8 MB PSRAM, low power MCU
Tiny-KNN	240 MHz	300 KB	0.30 – 0.40 W	Suitable for small dataset classification
Tiny-RF	240 MHz	400 – 600 KB	0.35 – 0.45 W	Works with small tree depth
Tiny-SVM	240 MHz	200 – 300 KB	0.30 – 0.40 W	Efficient with linear kernel
Standard ML models (for PC/GPU)				
DNN	2 GHz	2 GB	50 W	Not supported
KNN	2 GHz	2 GB	60 W	Not supported
RF	3.5 GHz	4 GB	70 W	Not supported
SVM	2 GHz	2 GB	60 W	Not supported

Tab. 2. Hardware specifications and timing parameters.

Parameter	Description	Value
Hardware specifications		
P_{SENSE}	Power consumption for sensing (Grove acoustic sensor)	0.035 W
$P_{PROCESS}$	Power consumption for processing (XIAO ESP32S3)	0.35 W
P_{TX}	Power consumption for transmission (XBee)	1.25 W
P_{RX}	Power consumption for receiving (XBee)	1.2 W
Timing parameters		
T_{SENSE}	Sensing time (duration of audio file)	5.0 s
$T_{PROCESS\_SENSOR}$	Processing time (XIAO ESP32S3)	0.2 s
$T_{PROCESS\_ROUTER}$	Processing time (router node – XBee)	0.1 s
T_{TX}	Transmission time (XBee, 431 KB file)	0.275 s
T_{RX}	Receiving time (XBee, 431 KB file)	0.275 s

5.2. Performance Evaluation and Comparative Analysis

Figure 7 presents a comparative analysis of five consecutive event listening instances in four pervasive sensor zones, focusing on the detection of four specific acoustic events with a detection accuracy rate exceeding 0.7. The analysis includes four different TinyML models and a baseline scenario without TinyML, where all events are detected and forwarded to the base station. In all cases, the figure demonstrates energy efficiency gains while maintaining a good event detection accuracy rate. Furthermore, it compares energy efficiency across different TinyML models to identify the most optimal model for the ESC-50 dataset. The results indicate that the RF algorithm achieves the best performance, with an energy efficiency of approximately 40%.

Performance evaluation and comparative analysis were also performed to assess the effectiveness of different TinyML models deployed at sensor nodes and to compare it with the effectiveness of standard models running on high-performance computers. Figures 8 and 9 illustrate that TinyML models, despite operating in resource-constrained environments, achieve performance levels comparable to those of standard models

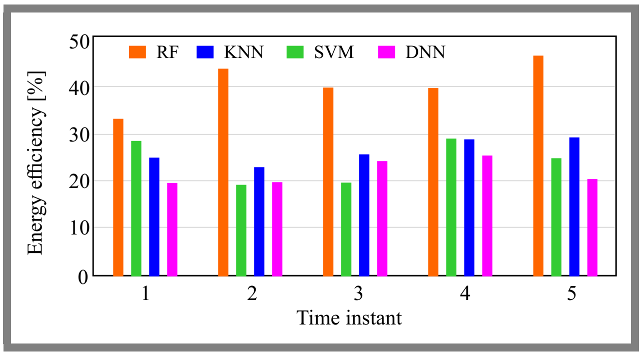

Fig. 7. Energy efficiency of different TinyML models on Seeed XIAO ESP32S3.

used in advanced systems. The evaluation considered multiple classifications of acoustic events, with models such as SVM, RF, KNN and DNN being tested at the sensor nodes. The results demonstrate that among the TinyML models evaluated, the RF model exhibits superior performance in terms of classification accuracy, while simultaneously maintaining good energy efficiency. Based on these findings, the RF model is recommended for implementation at the sensor nodes, as

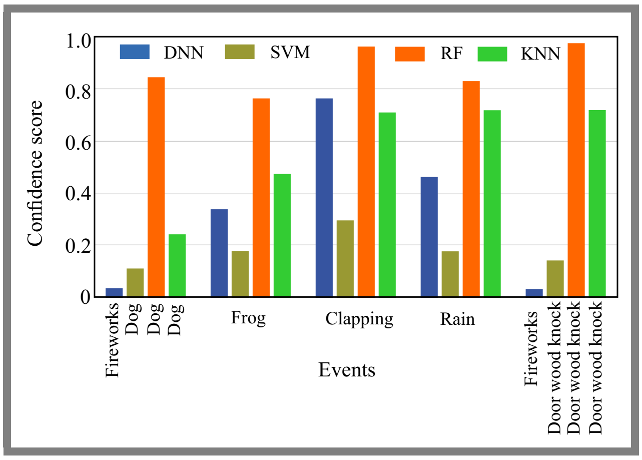

Fig. 8. Event detection accuracy for different TinyML models on Seeed XIAO ESP32S3i.

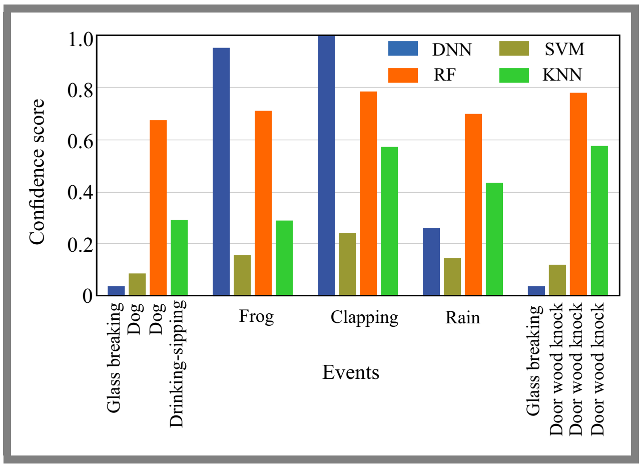

Fig. 9. Event detection accuracy for different standard ML models at the base station.

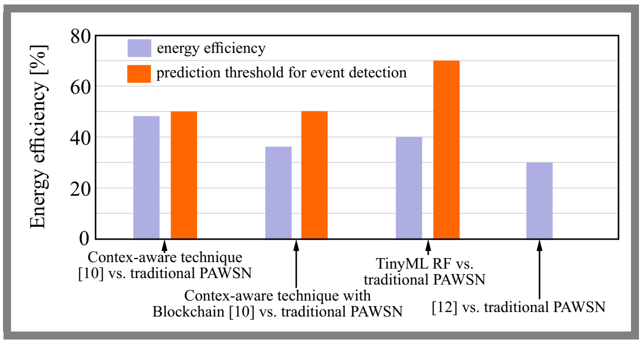

Fig. 10. Comparison of energy efficiency with context-aware model.

it ensures reliable and accurate event detection with minimal computational overhead.

Performance is also compared with the findings of the work on blockchain-based acoustic data integration for PAWSN [10], where the context-aware event detection technique was used to reduce transmission cost by decreasing the number of receiving and transmitting operations. This technique replaces these context-sensitive semantic sensor nodes with TinyML-based AED sensor nodes, enabling more accurate detection of acoustic events while simultaneously reducing the significant amount of acoustic data transmission, and thus contributing to greater energy efficiency. Figure 10 shows that energy efficiency performance of the two models remains almost the same, while the event prediction threshold is set to 0.5 in [10] and 0.7 in the present model, highlighting the more accurate event detection process of the proposed model. The comparison also includes a blockchain-enhanced context-aware technique [12] and a traditional WSN-based approach, further demonstrating the trade-offs between energy efficiency and event detection accuracy.

6. Conclusion and Future Scope

This work demonstrates the effectiveness of TinyML-driven sensor nodes for energy-efficient acoustic event detection in pervasive wireless sensor networks (PAWSNs). By relying on lightweight machine learning models, the proposed approach significantly reduces data transmission, optimizes energy consumption and maintains a high classification accuracy level.

Comparative analysis confirms that TinyML models achieve performance levels comparable to those of standard high-performance computing models, making them a viable alternative for real-time edge computing applications. Among the models evaluated, the random forest algorithm is efficient, achieving approximately 40% energy savings while ensuring reliable event detection.

The findings highlight the potential of TinyML to improve the longevity and sustainability of sensor networks deployed in resource-constrained environments. Future work can improve TinyML adaptability with real-time updates and self-learning while ensuring scalability across industries, wildlife, and smart cities, with energy harvesting deployed to extend network longevity.

References

[1] S. Das and U. Mondal, "Acoustic Data Acquisition and Integration for Semantic Organization of Sentimental Data and Analysis in a PWSN", *Multimedia Tools and Applications*, 2024 (https://doi.org/10.1007/s11042-024-20229-4).

[2] S. Das and U. Mondal, "Pilot Agent Implied Efficient Data Communication in Pervasive Acoustic Wireless Sensor Network", *Telecommunication Systems*, vol. 88, art. no. 50, 2025 (https://doi.org/10.1007/s11235-025-01281-3).

[3] P. Andrade *et al.*, "A TinyML Soft-sensor Approach for Low-cost Detection and Monitoring of Vehicular Emissions", *Sensors*, vol. 22, art. no. 3838, 2022 (https://doi.org/10.3390/s22103838).

[4] P. Yadav, "Advancements in Machine Learning in Sensor Systems: Insights from Sensys-ML and TinyML Communities", *2024 IEEE 3rd Workshop on Machine Learning on Edge in Sensor Systems (SenSys-ML)*, Hong-Kong, 2024 (https://doi.org/10.1109/SenSys-ML62579.2024.00009).

[5] A. Sabovic *et al.*, "Towards Energy-aware TinyML on Battery-less IoT Devices", *Internet of Things*, vol. 22, art. no. 100736, 2023 (https://doi.org/10.1016/j.iot.2023.100736).

[6] H.H. Ahmed, Z. Ahmed, T. Choden, and N. Chaudhary, "TinyML for Emotion Detection in Voice Signals: Evaluating and Proposing Algorithms for IoT Wearable Devices, *Thesis*, Brac University, 2024 [Online] Available: http://hdl.handle.net/10361/24346.

[7] S. Hammad, D. Iskandaryan, and S. Trilles, "An Unsupervised TinyML Approach Applied to the Detection of Urban Noise Anomalies un-

der the Smart Cities Environment", *Internet of Things*, vol. 23, art. no. 100848, 2023 (https://doi.org/10.1016/j.iot.2023.100848).

[8] Z. Huang *et al.*, "TinyChirp: Bird Song Recognition Using TinyML Models on Low-power Wireless Acoustic Sensors", *2024 IEEE 5th International Symposium on The Internet of Sounds (IS2)*, Erlangen, Germany, 2024 (https://doi.org/10.1109/IS262782.2024.10704131).

[9] A. Elhanashi, P. Dini, S. Saponara, and Q. Zheng, "Advancements in TinyML: Applications, Limitations, and Impact on IoT Devices", *Electronics*, vol. 13, art. no. 3562, 2024 (https://doi.org/10.3390/electronics13173562).

[10] S. Das and U. Mondal, "Energy Efficient Acoustic Sensor Data Integration in Hybrid Mode Operated Pervasive Wireless Sensor Network", *Telecommunication Systems*, vol. 87, pp. 61–72, 2024 (https://doi.org/10.1007/s11235-024-01165-y).

[11] G. Cerutti, R. Prasad, A. Brutti, and E. Farella, "Compact Recurrent Neural Networks for Acoustic Event Detection on Low-energy Low-complexity Platforms", *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, pp. 654–664, 2020 (https://doi.org/10.1109/JSTSP.2020.2969775).

[12] M. Andhare *et al.*, "Design and Implementation of Wireless Sensor Network for Environmental Monitoring", *International Journal of Health Sciences*, vol. 6, pp. 3158–3169, 2022 (https://doi.org/10.53730/ijhs.v6nS4.9085).

[13] G. Cerutti *et al.*, "Sound Event Detection with Binary Neural Networks on Tightly Power-constrained IoT Devices", *Proc. of the ACM/IEEE International Symposium on Low Power Electronics and Design*, pp. 19–24, 2020 (https://doi.org/10.1145/3370748.3406588).

[14] M. Antonini, M. Pincheira, M. Vecchio, and F. Antonelli, "An Adaptable and Unsupervised TinyML Anomaly Detection System for Extreme Industrial Environments", *Sensors*, vol. 23, art. no. 2344, 2023 (https://doi.org/10.3390/s23042344).

[15] S. Githu, "Detecting Worker Accidents with Audio Classification – Syntiant TinyML", Edge Impulse, [Online] Available: https://docs.edgeimpulse.com/experts/audio-projects/detecting-worker-accidents-syntiant-tinyml.

[16] Q. Wu, P. Sun, and A. Boukerche, "An Energy-efficient UAV-based Data Aggregation Protocol in Wireless Sensor Networks", *Proc. of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Application*, pp. 34–40, 2018 (https://doi.org/10.1145/3272036.3272047).

[17] J. Yan, X. Yang, X. Luo, and C. Chen, "Energy-efficient Data Collection over AUV-assisted Underwater Acoustic Sensor Network", *IEEE Systems Journal*, vol. 12, pp. 3519–3530, 2018 (https://doi.org/10.1109/JSYST.2017.2789283).

[18] Z. Zhou, S. Zhou, J.-H. Cui, and S. Cui, "Energy-efficient Cooperative Communication Based on Power Control and Selective Single-relay in Wireless Sensor Networks", *IEEE Transactions on Wireless Communications*, vol. 7, pp. 3066–3078, 2008 (https://doi.org/10.1109/TWC.2008.061097).

[19] C. Chen *et al.*, "Ubiquitous Monitoring for Industrial Cyber-physical Systems over Relay-assisted Wireless Sensor Networks", *IEEE Transactions on Emerging Topics in Computing*, vol. 3, pp. 352-362, 2015 (https://doi.org/10.1109/TETC.2014.2386615).

[20] S. Heydari and Q.H. Mahmoud, "Tiny Machine Learning and On-device Inference: A Survey of Applications, Challenges, and Future Directions", *Sensors*, vol. 25, art. no. 3191, 2025 (https://doi.org/10.3390/s25103191).

[21] L. Capogrosso *et al.*, "A Machine Learning-oriented Survey on Tiny Machine Learning", *IEEE Access*, vol. 12, pp. 23406–23426, 2024 (https://doi.org/10.1109/ACCESS.2024.3365349).

Bibek B. Roy, M.Sc.
Department of Computer Science
https://orcid.org/0009-0001-6699-1551
E-mail: roybibek16@gmail.com
Vidyasagar University, Midnapore, WB, India
https://www.vidyasagar.ac.in

Sushovan Das, Ph.D.
Department of CSE
https://orcid.org/0000-0003-2759-3902
E-mail: das.sushovan@gmail.com
College of Engineering & Management, Kolaghat, WB, India
https://www.cemkolaghat.in

Uttam Kr. Mondal, Ph.D.
Department of Computer Science
https://orcid.org/0000-0002-7807-3002
E-mail: uttam_ku_82@yahoo.co.in
Vidyasagar University, Midnapore, WB, India
https://www.vidyasagar.ac.in