
Smart Sensor-Based Hive Monitoring System Using IoT: A Distributed Architecture for Real-Time Beehive Health Assessment

Aditya Sehrawat¹, S.D. Samantaray¹

¹*Department of Computer Engineering, College of Technology, GBPUA&T, Pantnagar, 263145, India*

Abstract - Traditional beekeeping relies on periodic manual hive inspections, which are labor-intensive, disruptive to colonies, and unable to detect sudden changes between visits. This paper presents a low-cost, IoT-based Smart Sensor-Based Hive Monitoring System for continuous, non-invasive monitoring of honeybee colony health. The system uses three independent ESP32 nodes: a BME280 sensor for temperature, humidity, and atmospheric pressure; an INMP441 MEMS microphone for acoustic activity using on-device RMS computation; and an HX711 amplifier with a load cell for hive weight measurement. Sensor data are securely transmitted via MQTT to a HiveMQ Cloud broker, where a Python application logs timestamped readings to Excel and generates automated HTML email alerts when predefined thresholds are exceeded. A Raspberry Pi 4 with Camera Module V2 captures scheduled videos of the hive entrance to monitor foraging activity. Field testing, including a real bee colony session, confirmed reliable environmental sensing, clear differentiation of ambient, mechanical, and bee acoustic signatures, accurate weight measurement with low calibration error, and complete data logging without missing or duplicate records. The results demonstrate that the proposed modular, cloud-connected system provides a practical, scalable, and cost-effective solution for continuous hive monitoring and informed beekeeping.

Key Words: IoT, precision apiculture, ESP32, MQTT, hive monitoring, acoustic sensing.

1. INTRODUCTION

Honeybees are among the most efficient pollinators and play a central role in agricultural productivity and ecological balance. The Food and Agriculture Organization estimates that roughly 75% of the world's principal food crops depend on animal pollination, with pollination services valued at over USD 235-577 billion annually worldwide (Potts et al., 2010). Despite this importance, honeybee populations continue to decline owing to pesticide exposure, climate change, habitat loss and disease, placing colony health monitoring at the center of sustainable beekeeping and food security efforts.

Most beekeeping operations still rely on manual hive inspection, in which a beekeeper opens the hive every one to two weeks to check the brood pattern, food stores and queen status. This approach is time-consuming, disturbs the colony and, more importantly, cannot capture rapid changes such as queen loss, heat stress or swarming that can occur between two routine visits. Internet of Things (IoT) based sensing offers a reliable alternative (Ceccho et al., 2020; Andrijevic et al., 2022): continuous, low-disturbance data collection that can flag anomalies as they happen rather than at the next scheduled inspection.

A honeybee colony behaves as a coordinated superorganism rather than a collection of independent insects, so its condition is best inferred from externally observable, colony-level signals rather than from any single bee. Internal hive temperature and humidity reflect the brood-nest microclimate that the colony actively regulates; hive weight reflects nectar intake and food reserves; acoustic activity reflects wing-fanning and general colony behavior; and entrance activity reflects foraging intensity. None of these parameters alone is conclusive but monitored together they allow abnormal conditions to be

confirmed with much greater confidence than any single reading.

Beyond the immediate motivation, traditional beekeeping also carries structural challenges that a monitoring layer alone cannot fix but can help mitigate: permanent-comb hives limit inspection, poor infrastructure limits market access, and habitat loss and pesticide exposure shrink the foraging range available to a colony. IoT-based environmental monitoring has already proven valuable (Hamza et al., 2023) across agriculture more broadly, allowing continuous sensing to replace intermittent manual checks and enabling cloud-based storage, historical trend analysis and remote decision-making with reduced manpower. Extending this same approach to apiculture, in the form of a smart beehive monitoring system (Dsouza et al., 2023; Liyanage et al., 2024), allows temperature, humidity, pressure, weight and acoustic activity to be logged continuously and combined with entrance-activity video, giving a considerably more complete picture of colony condition than periodic manual inspection alone.

A Langstroth hive box was chosen as the physical platform for this study because it is the most widely used hive design in both commercial and small-scale beekeeping, its stacked, movable-frame construction allows inspection without disturbing the whole colony, and its flat base and accessible frame structure made it convenient to mount weight, environmental and acoustic sensors with minimal disruption to the bees. Using a widely adopted hive design also makes the results easier to compare against other IoT-based hive-monitoring studies.

This work presents the design, implementation and field evaluation of a distributed, IoT-based hive monitoring system built around a Langstroth hive box. The system was designed to address several gaps observed in prior work (Kviesis et al., 2023; Moreno et al., 2025): reliance on a single centralized data-collection hub creating a single point of failure; use of

non-standard, higher-complexity communication protocols where simpler options would suffice for a single-apiary deployment; acoustic sensing treated as a secondary parameter rather than a continuously logged one; dependence on third-party dashboards rather than an independent, human-readable data and alert pipeline; and camera-based monitoring rarely operated as an independent, scheduled subsystem alongside environmental sensing.

2. OBJECTIVES

- Design and develop an IoT-based smart beehive monitoring system for continuous, real-time tracking of hive conditions.
- Monitor temperature, humidity, atmospheric pressure, hive weight and acoustic activity using a distributed, single-sensor-per-node architecture.
- Integrate a Raspberry Pi with a camera module for scheduled visual monitoring of bee activity at the hive entrance.
- Transmit sensor data wirelessly to a cloud MQTT broker for secure, remote access.
- Record and store time-stamped sensor data and generate automated e-mail alerts when abnormal conditions are detected, reducing the need for manual inspection.

3. METHODOLOGY

3.1 System architecture

This system follows a distributed sensing architecture built around a single Langstroth hive box. Three independent ESP32 microcontroller nodes were used, each dedicated to one sensing modality, and a separate Raspberry Pi 4 handled visual monitoring. Each ESP32 read its sensor, performed any necessary on-device processing, and published a compact JSON message every seven seconds over Wi-Fi to a cloud-hosted MQTT broker (HiveMQ Cloud) using a TLS-secured connection on port 8883. A Python script running continuously on a laptop subscribed to a single wildcard topic covering all three nodes, wrote every incoming reading to a time-stamped Microsoft Excel sheet using openpyxl, and issued an automated HTML e-mail alert through Gmail SMTP whenever a reading crossed a pre-defined threshold. This design avoided a single centralized hub for the sensor nodes: a fault in one node did not affect the others, and lightweight MQTT/Wi-Fi kept the hardware and configuration complexity low compared with dedicated long-range radio protocols, which are better suited to multi-apiary deployments spanning larger areas.

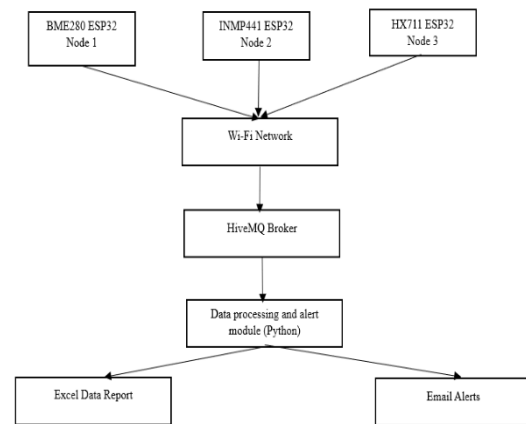


Fig -1: Data Flow Diagram for Sensors

Two independent data paths operate within the system. The sensor path begins at the three ESP32 nodes, each reading its dedicated sensor and publishing a JSON payload to the HiveMQ broker; the Python subscriber then produces two outputs for every incoming reading — an Excel log entry and, when applicable, an e-mail alert. The video path is entirely separate: the Raspberry Pi records and stores clips locally on its own fixed schedule and these are retrieved afterward over Wi-Fi, so a delay or fault in the sensor path has no effect on video capture and vice versa.

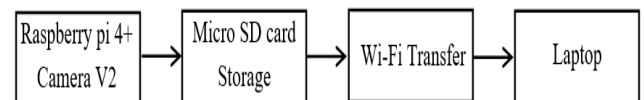


Fig -2: Data Flow Diagram for Video Capture

Processing sensor data locally, on the ESP32 itself, before transmission — rather than streaming raw sensor output to the cloud — was a deliberate design choice applied consistently across all three nodes: it reduced the volume of data sent over MQTT, lowered the load on the broker, and meant the data arriving at the Python subscriber was already in a directly usable form, simplifying the receiving-end logic.

3.2 Hardware Components

Table 1 lists the hardware used. The BME280 (Bosch Sensortec) measured internal temperature, relative humidity and barometric pressure over I2C (Bosch Sensortec GmbH, 2022) and was inserted through a small opening into the hive interior. The INMP441 MEMS digital microphone captured acoustic activity over I2S (TDK InvenSense, 2021); rather than streaming raw audio, the ESP32 computed the Root Mean Square (RMS) amplitude of each sample buffer on-device and published only this single representative value per cycle, since transmitting raw audio over MQTT in real time was impractical. A 40 kg single-point load cell with an HX711 amplifier, sandwiched between

two wooden platforms beneath the hive, measured total hive weight. All three ESP32 nodes were housed in separate IP65-rated enclosures on perfboard assemblies for outdoor durability, and a Raspberry Pi 4 with Camera Module V2, in a 3D-printed enclosure with an adjustable camera arm, was positioned to view the hive entrance. Four 10,000 mAh power banks without automatic low-current shutoff supplied all units, enabling mains-independent field deployment.

Component	Qty	Purpose
ESP32 Dev. Board	3	Sensor acquisition and MQTT publishing
BME280 Sensor	1	Temperature, humidity, pressure
INMP441 MEMS Mic.	1	Acoustic activity (RMS)
HX711 + Load Cell (40 kg)	1	Hive weight
Raspberry Pi 4 Model B	1	Camera control and video capture
Camera Module V2	1	Hive entrance video
IP65 Enclosures	3	Weatherproofing of ESP32 nodes
Power Banks (10,000 mAh)	4	Portable field power supply

Table -1: Hardware Components



Fig -3: Prototype of the proposed IoT-based smart hive monitoring system.

3.3 Sensor Calibration and Verification

Each sensor was individually verified before deployment. The BME280 was cross-checked against a reference thermometer and hygrometer and found to be within its rated $\pm 1^\circ\text{C}$ and $\pm 3\%$ RH accuracy. The INMP441 was verified by observing a stable, low RMS baseline in silence and a sharp, repeatable RMS increase when a buzzer was placed near the microphone. The HX711 load cell was calibrated by first taring the platform and then applying a known reference weight to compute a calibration factor (-28499 , the negative

sign reflecting the compression-mount orientation), which was embedded in the firmware; nine independent reference weights spanning 2–16 kg were subsequently used to verify the calibration, with results reported in Section 4.3. All sensor calibration and verification procedures were carried out using the Arduino IDE.

3.4 Cloud Communication and Data Pipeline

MQTT (Message Queuing Telemetry Transport) was chosen over HTTP polling (Andrijevic et al.,2022) because it maintains a persistent, lightweight publish-subscribe connection well suited to small, frequent IoT payloads, rather than requiring the client to repeatedly poll for new data. QoS 0 was used, as occasional loss of a single periodic reading does not materially affect trend-based hive monitoring. Each ESP32 node published to its own MQTT topic (e.g., apiary/hive1/environment,apiary/hive1/sound, apiary/hive1/weight), and the Python subscriber used a single wildcard subscription to receive all three streams, appending every reading to the Excel log with a locally generated timestamp and checking it against threshold rules before deciding whether to trigger an e-mail alert.

3.5 Visual Monitoring

The Raspberry Pi ran an independent cron-scheduled script that recorded a one-minute video clip every five minutes, balancing entrance-activity coverage against onboard microSD storage. Recorded clips were named with an embedded timestamp and transferred wirelessly to a laptop using scp, avoiding the need to physically remove the storage card in the field.

3.6 Power Supply and Data Security

The system was powered entirely by portable 10,000 mAh power banks (one per ESP32 node and one for the Raspberry Pi) rather than mains electricity, allowing field deployment away from fixed power infrastructure, similar to real apiary conditions. Power banks without automatic low-current shutoff were deliberately selected, since the intermittent, low average current draw of an idling ESP32 or Raspberry Pi could otherwise trigger a standard power bank's auto-off protection and cause unplanned node disconnection. Because each node had its own independent power source, a fault or depletion in one power bank did not affect the other nodes, improving overall fault tolerance.

The design also considered basic data privacy and communication security. No personally identifiable information was collected at any stage — only sensor readings (temperature, humidity, pressure, RMS sound level, weight) and their timestamps were gathered and transmitted. All MQTT traffic between the ESP32 nodes, the broker and the Python subscriber was encrypted using TLS on port 8883, and username/password authentication was enforced for every client connecting to the broker, preventing unauthorized publishing or subscribing.

4. Results and Discussion

The system was field-tested on a working Langstroth hive over a series of controlled sessions designed to isolate the response of each sensing subsystem: diurnal environmental sessions (morning, afternoon and night, repeated daily over seven days, 22-28 June 2026), a single-session real-bee-colony exposure, a tree-shaded microclimate session, four one-hour acoustic test conditions, a nine-point load-cell calibration, and dedicated tests of the camera and data-logging subsystems.

4.1 BME 280 Environmental Sensor

Table 2 summarizes the mean environmental readings across all five BME280 test conditions, each based on continuous 7-8 second interval sampling. Temperature followed the expected diurnal pattern, peaking in the afternoon (37.57 °C) and reaching its minimum in the morning (29.6 °C), a difference of 7.97 °C between the two extremes, while atmospheric pressure stayed within a narrow 973.22-973.80 hPa band across every condition — less than 0.6 hPa of total spread — confirming that pressure is governed by broader weather patterns rather than colony or hive-box activity. Humidity moved inversely with temperature across the three diurnal sessions, from 63.30% RH in the cooler morning down to 52.58% RH in the hotter afternoon. The most notable finding was in the real-bee-colony session, recorded in the afternoon time slot: mean humidity reached 62.04% RH, 9.46 percentage points above the 52.58% RH afternoon ambient baseline, while the internal temperature (34.02 °C) sat well below the 37.57 °C afternoon baseline and closer to the night-session value (32.29 °C). This indicates that the colony was actively regulating its internal microclimate — through wing-fanning, water-gathering and clustering behavior — rather than simply tracking outside conditions, consistent with the well-documented thermoregulatory behavior of honeybee colonies (Potts et al.,2010). The tree-surrounded session recorded a moderate temperature (30.03 °C) and humidity (54.94% RH), confirming the expected cooling effect of natural shade and offering a useful data point for future hive-siting decisions.

Condition	Temp. (°C)	Humidity (%RH)	Pressure (hPa)
Morning (7-day)	29.60	63.30	973.80
Afternoon (7-day)	37.57	52.58	973.61
Night (7-day)	32.29	58.59	973.22
Real bee colony	34.02	62.04	973.66
Tree-surrounded	30.03	54.94	973.53

Table -2: Mean Environmental Parameters Across Test

Conditions

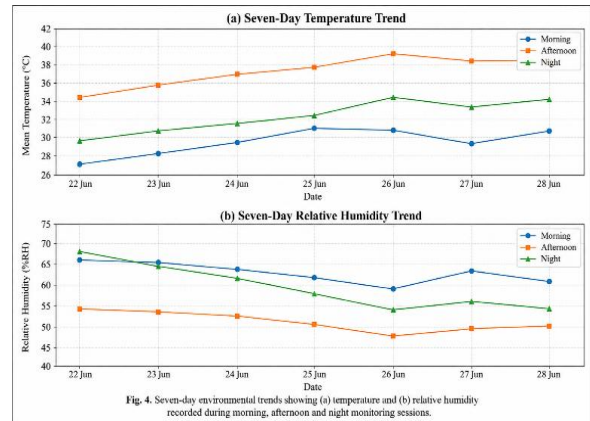


Fig -4: Seven-day variation in (a) temperature and (b) relative humidity recorded during morning, afternoon, and night monitoring sessions.

4.2 INMP441 Acoustic Sensor

Table 3 summarizes the RMS statistics obtained under four acoustic conditions. The ordering silence (175.59) < fan noise (204.45) < real bee colony (347.09) < music (585.31) matched the expected physical characteristics of each source. The real-bee-colony RMS was nearly double the silence baseline and clearly separated from the fan-noise range, while its standard deviation (58.10) was far lower than that of music (210.55), indicating that an undisturbed colony produces a comparatively steady acoustic signature — a useful property for a future threshold-based swarm or disturbance alert (Cecchi et al.,2020).

Condition	Mean RMS	Std. Dev.	Range
Silence	175.59	32.90	117 - 235
Fan noise	204.45	45.30	150 - 312
Real bee colony	347.09	58.10	250 - 449
Music	585.31	210.55	223 - 993

Table -3: RMS Statistics Under Different Acoustic Conditions

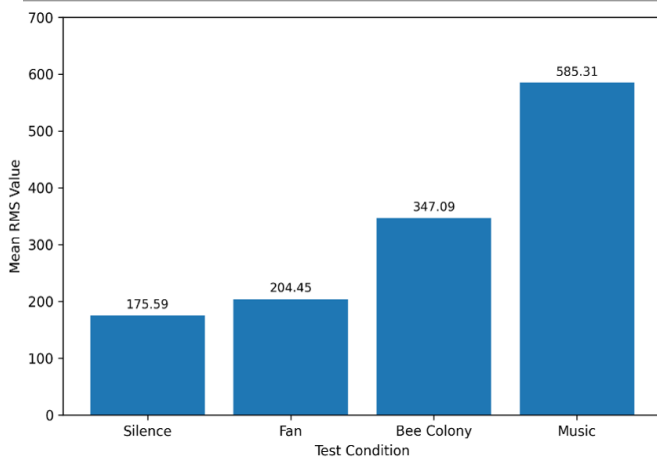


Fig -5: Acoustic RMS Comparison

4.3 HX711 Load Cell

Table 4 reports the accuracy of the HX711 load-cell subsystem across nine reference weights from 2 kg to 16 kg. The mean absolute percentage error was 1.24%, with the largest relative errors occurring at the lowest reference weights (+3.00% at 2 kg) and errors below 1% consistently observed in the 10-16 kg range that is most representative of real hive weights. A linear regression of measured against known weight gave a near-unity slope (0.996) and a negligible intercept (0.101 kg) with $RZ = 0.9999$, confirming excellent linearity across the calibrated range.

Metric	Value
Mean Absolute Error (MAE)	0.068 kg
Root Mean Square Error (RMSE)	0.085 kg
Mean Absolute Percentage Error (MAPE)	1.24%
Linear fit	$y = 0.996x + 0.101$, $RZ = 0.9999$

Table -4: Statistical Accuracy of the HX711 Load Cell (2-16 kg)

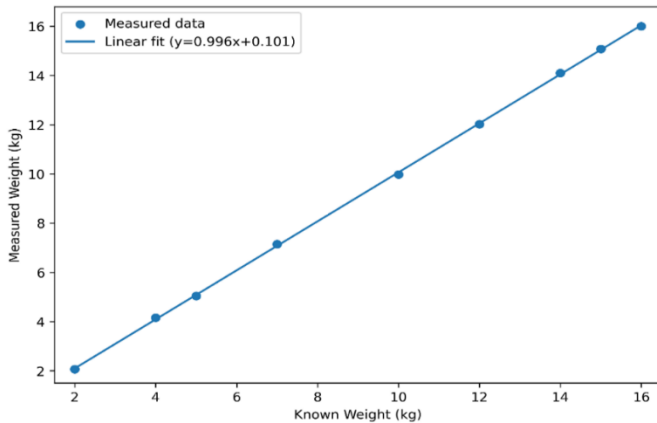


Fig -6: Load Cell Calibration Curve

4.4 Data Logging, Alerting and Camera Performance

Table 5 summarizes performance of the Excel-based logging pipeline during a fully connected 30-minute session with all three ESP32 nodes active. All 205 log entries were complete, with zero missing cells and zero duplicate entries, at an average logging rate of 6.6 entries per minute against a design target of roughly seven. Cross-checking recorded e-mail alerts against the corresponding Excel entries showed no mismatch in timing or value, confirming that the alerting and logging paths read from the same underlying MQTT stream consistently.

Metric	Value
Total log entries	205
Session duration	30 minutes
Data completeness	100%
Duplicate entries	0
Avg. entries per minute	6.6

Table -5: Excel-Based Data Logging Performance (30-min Session)

The Raspberry Pi camera subsystem recorded clips at 5 minutes 1 second intervals against a 5-minute target, indicating negligible scheduling drift over repeated cycles. Average clip size was approximately 11.5 MB, giving an estimated storage capacity of about 6.3 days of continuous recording on the available 21 GB of usable microSD space (32 GB card, roughly 8 GB consumed by the OS and project scripts), and wireless transfer of each clip completed in about 2 seconds with no corrupted files. Sample frames from a real hive-entrance session confirmed clear visibility of individual bees under natural lighting, validating both the placement and the field of view of the camera assembly.

Based on typical component current draw, the individual ESP32 sensor nodes were estimated to run for approximately 43 hours on a single 10,000 mAh power bank, while the more power-hungry Raspberry Pi camera unit was estimated at approximately 11 hours; no voltage drops, disconnections or performance degradation were observed in any node during the testing sessions. In a longer field deployment this suggests the sensor nodes could reasonably run for a day or two between charges, while the camera unit would need more frequent charging or a larger power bank.

4.5 Discussion

Taken as a whole, the results across all subsystems support the central premise of the study: that a fully distributed, single-sensor-per-node architecture communicating over lightweight MQTT can deliver environmental, acoustic,

weight and visual monitoring with an accuracy and reliability comparable to more centralized or protocol-heavy designs, while keeping hardware and configuration complexity low enough for a single-apiary or research deployment. In particular, the clear separation between the real-bee-colony acoustic and humidity signatures and their respective ambient baselines suggests that both metrics are strong, low-cost candidates for future automated anomaly-detection rules (e.g., a sudden RMS drop toward the silence baseline as an early indicator of colony loss or swarming) (Bilik et al., 2024), while the sub-2% weight-measurement error over the practically relevant 10–16 kg range confirms the system is fit for tracking honey accumulation and food-store trends over time.

5. Conclusion

This work presented the design and field evaluation of a low-cost, distributed IoT architecture for continuous, non-invasive honeybee hive monitoring. Rather than relying on a single centralized hub, three independent ESP32 nodes each handled one sensing modality — temperature/humidity/pressure, acoustic RMS and weight — publishing securely to a cloud MQTT broker, while a separate Raspberry Pi captured scheduled entrance video. A Python-based pipeline logged all readings to a structured, human-readable Excel record and issued automated e-mail alerts without dependence on a third-party dashboard.

Field results confirmed that each subsystem performed reliably under real conditions: the BME280 captured both diurnal trends and a measurable colony-driven humidity increase; the INMP441 clearly distinguished ambient, mechanical and bee-colony acoustic signatures using only an on-device RMS metric; the HX711 load cell achieved sub-2% mean error with near-perfect linearity; and the logging and alerting pipeline achieved complete, duplicate-free data capture. These results support the feasibility of a modular, self-contained and cost-effective alternative to centralized or dashboard-dependent hive monitoring systems for small-scale and research apiary deployments.

Future work can extend this system with machine-learning-based analysis of the acoustic and weight time series to preemptively flag swarming or disease risk, additional gas sensing for early pest or disease detection, solar-based power to remove the need for periodic battery replacement, multi-hive scaling, and GPS-based tracking for mobile hive deployments

ACKNOWLEDGEMENT

The authors thank the Department of Computer Engineering and the Honeybee Research Centre, G. B. Pant University of Agriculture and Technology, Pantnagar, for facilities and support extended during data collection for this study.

REFERENCES

1. Andrijević, N., Urošević, V., Arsić, B., Herceg, D. and

- Savić, B. 2022. IoT monitoring and prediction modeling of honeybee activity with alarm. *Electronics*, 11(5): 783.
2. Bilik, S., Zemeik, T., Kratochvila, L., Rikanek, D., Richter, M., Zambanini, S. and Horak, K. 2024. Machine learning and computer vision techniques in continuous beehive monitoring applications: A survey. *Comput. Electron. Agric.*, 217: 108560.
3. Bosch Sensortec GmbH. 2022. BME280 Combined Humidity, Pressure and Temperature Sensor Datasheet. Bosch Sensortec GmbH, Germany.
4. Cecchi S., Spinsante, S., Terenzi, A. and Orcioni, S. 2020. A smart sensor-based measurement system for advanced bee hive monitoring. *Sensors*, 20(9): 2726.
5. Dsouza, A., Aditya, P. and Hegde, S. 2023. HiveLink, an IoT based smart bee hive monitoring system. *arXiv preprint, arXiv:2309.12054*.
6. Espressif Systems. 2023. ESP32 Series Datasheet. Espressif Systems, Shanghai, China.
7. Hamza, A. S., Tashakkori, R., Underwood, B., O'Brien, W. and Campell, C. 2023. BeeLive: The IoT platform of Beemon monitoring and alerting system for beehives. *Smart Agric. Technol.*, 5: 100331.
8. Kviesis, A., Komasilovs, V., Ozols, N. and Zacepins, A. 2023. Bee colony remote monitoring based on IoT using ESP-NOW protocol. *PeerJ Comput. Sci.*, 9: e1363.
9. Liyanage, N., Attanayaka, C., Perera, T., Neilkumara, D., Bandara, I. S. and Chandrasiri, L. 2024. IoT-based smart beehive monitoring system. In: 6th Int. Conf. on Advancements in Computing (ICAC), Colombo, Sri Lanka. pp. 247-252.
10. Moreno, C. A., Montañez, T. F., Cortés, S., Paz Penagos, H. and Hernandez, S. V. 2025. Real-time monitoring system of a beehive through IoT. *SSRN Electron. J.*
11. Potts, S. G., Biesmeijer, J. C., Kremen, C., Neumann, P., Schweiger, O. and Kunin, W. E. 2010. Global pollinator declines: Trends, impacts and drivers. *Trends in Ecology & Evolution*, 25(6): 345-353.
12. TDK InvenSense. 2021. INMP441 MEMS Microphone Datasheet. TDK InvenSense Inc., USA.
13. Zaman, A. and Dorin, A. 2023. A framework for better sensor-based beehive health monitoring. *Comput. Electron. Agric.*, 206:
