

# A LoRa and MQTT-Based Monitoring System for Internal and External Beehive Conditions

1<sup>st</sup> Minju Jeon  
Division of AI Computer Science and Engineering  
Kyonggi University  
Suwon, South Korea  
mingmingmon@kyonggi.ac.kr

2<sup>nd</sup> Jiyun Kim  
College of Information Science  
Hallym University  
Chuncheon, South Korea  
20225141@hallym.ac.kr

3<sup>rd</sup> Sewon Kim  
Division of Computer Science and Engineering  
Jeonbuk National University  
Jeonju, South Korea  
andimsewon@jbnu.ac.kr

4<sup>th</sup> Seongmin Park  
Division of Computer Science and Engineering  
Kyonggi University  
Suwon, South Korea  
duke7272@kyonggi.ac.kr

5<sup>th</sup> Bo Zhang  
Polytechnic Institute  
Purdue University  
Indiana, United States  
zhan5055@purdue.edu

6<sup>th</sup> Anthony H. Smith  
Computer and Information Technology  
Purdue University  
West Lafayette, IN, USA  
ahsmith@purdue.edu

**Abstract**—Bees play a critical role in agriculture and global food production, yet their populations are declining, driven by habitat loss, pesticide exposure, and climate change. Precision beekeeping enables real-time environmental monitoring, complementing traditional hive management practices. Smart beehives integrate embedded sensors to monitor internal environmental conditions such as temperature and humidity. However, existing solutions are limited by high cost, excessive weight, poor scalability, and a narrow focus on internal data alone. This paper proposes a cost-effective, scalable monitoring system that integrates LoRa and MQTT to address these challenges. The system uses Wi-Fi LoRa 32 (V3) modules in worker nodes to collect environmental data both inside and around the beehive. A cloud server and mobile application deliver real-time sensor data and alerts to beekeepers. Field experiments confirmed stable data collection and transmission over distances up to 110 meters in open settings, and up to 95 meters in obstructed environments. The system demonstrated robustness in both controlled and real-world deployments.

**Keywords**—Beekeeping, monitoring system, LoRa, MQTT

## I. INTRODUCTION

Bees play a vital role in both agriculture and global food security. Approximately 30% of global food production depends on bee pollination, with bees acting as the primary pollinators. However, this critical ecosystem service is under threat, as bee populations are highly vulnerable to environmental changes. Fluctuations in light and temperature have been shown to disrupt their behavior, physiology, and colony stability [1], [2]. To protect bee populations, continuous environmental monitoring is essential.

Recent developments have led to smart beehive systems equipped with environmental sensors. These systems collect data on temperature, humidity, and other internal hive conditions. However, existing solutions are constrained by key limitations in cost, size, and scope of sensing. First, most systems are expensive. Second, the hardware design is often heavy and difficult to deploy or maintain. Third, they lack modularity, making it difficult to scale or reconfigure the system for different hive layouts. Finally, most systems focus exclusively on internal conditions, overlooking external

environmental factors that significantly influence colony health [3].

To overcome these limitations, an improved system must be cost-effective, modular, and scalable, while supporting both internal and external monitoring. This paper proposes **WaggleNet**, a system that uses Wi-Fi LoRa 32 (V3) modules to implement worker nodes that collect environmental data. A master node, built with the same module, functions as a gateway between LoRa and MQTT. The collected sensor data is forwarded via MQTT to a cloud server, where it is stored in a structured database. A companion mobile application provides a real-time interface for visualizing hive conditions and environmental trends, enabling users to monitor system status remotely and efficiently.

This design helps reduce the overall cost of managing sensor devices. The Wi-Fi LoRa 32 (V3) module is more affordable than hardware used in conventional commercial systems, with a unit price of approximately \$10-15. Compared to conventional all-in-one systems, the proposed system deploys compact sensors both inside and around the hive. The compact size of the sensor modules allows for easy deployment across multiple hives without requiring structural modifications. Additional worker nodes can be placed as needed, and unnecessary ones can be removed or deactivated to adjust the monitoring scope. This plug-and-play approach simplifies both expansion and reduction of sensing network. Finally, the system captures both internal and external parameters, enabling more comprehensive assessment of environmental impacts on bee health.

The remainder of this paper is organized as follows: Section II provides background information relevant to the system design; Section III outlines the system requirements; Section IV describes the system architecture and role of each component; Section V presents experimental evaluation and field deployment results; Finally, Section VI concludes the paper and discusses future work.

## II. BACKGROUND

### A. IoT-based Beehive Monitoring Systems

Recent advancements on the Internet of Things (IoT) have driven substantial research into beehive monitoring systems. These systems typically include temperature, humidity, and light sensors placed inside or externally attached to the hive to capture environmental data. The collected data is used to assess colony health, detect anomalies early, and support bee productivity. However, existing literature predominantly focuses on internal hive conditions, often overlooking the influence of external environmental factors [4]. Because internal and external conditions are closely linked, ignoring external data limits accurate interpretation of hive dynamics. This study addresses the gap by simultaneously monitoring internal and external variables to identify cross-parameter correlations.

### B. LoRa-MQTT Integrated Communication Framework

Long Range (LoRa) technology is widely adopted for low-power, long-range data transmission. Based on Chirp Spread Spectrum (CSS) modulation, LoRa offers high energy efficiency for battery-powered IoT devices. LoRaWAN implementations, however, are constrained by limited processing throughput and scalability. To address these limitations, this work integrates LoRa with the Message Queuing Telemetry Transport (MQTT) protocol [5]. MQTT, a lightweight publish-subscribe messaging protocol, enhances real-time data handling and provides robust Quality of Service (QoS). The combination of LoRa's transmission range and MQTT's efficient data handling improves network reliability, especially in environments without LoRaWAN gateways or stable Wi-Fi access.

### C. ESP32-S3FN8 Microcontroller in Beehive Sensor Networks

The Heltec LoRa ESP32 V3 board used in this project integrates the ESP32-S3FN8 system-on-chip (SoC), a high-performance dual-core microcontroller from Espressif Systems. This chip couples Xtensa® 32-bit LX7 dual-core processing with integrated 2.4GHz Wi-Fi (802.11 b/g/n) and Bluetooth 5 (LE), operating at up to 240MHz and equipped with 512KB SRAM and 8MB of external flash memory. Rich digital and analog peripheral support—including multiple GPIO, ADC, DAC, PWM, SPI, and I2C lines—allows versatile sensor connections, while ultra-low-power sleep modes make it suitable for continuous field deployments in sensor networks. Hardware security features such as encryption and secure boot ensure data integrity in remote, real-world scenarios [19].

Unlike first-generation ESP32 variants, the S3FN8 model enhances AIoT capability with special instructions for machine learning and signal processing. Previous studies on the ESP32-S3 have highlighted its use not just in traditional

IoT applications—such as smart home control, energy monitoring, and wearable devices—but also as an embedded platform for edge computing tasks and visual sensor networks [20][21][22][23]. The S3 architecture's ample RAM and hardware vector instructions provide a foundation for advanced features like real-time anomaly detection or image processing, if needed in future smart beehive applications.

In the context of WaggleNet, the ESP32-S3FN8 serves as the backbone of both worker and master nodes, orchestrating reliable LoRa mesh communication and acting as an efficient gateway for Wi-Fi data transfer to cloud servers. The chip's flexible sleep and wake-up functions enable worker nodes to remain energy efficient—idling until scheduled data reporting or sensor-triggered events occur, such as detecting abnormal temperature or humidity levels within a hive. This platform's blend of low power consumption, multiple communication interfaces, and robust processing makes it ideal for scalable, modular, and remotely maintainable beehive monitoring in real agricultural environments.

## III. REQUIREMENT

**Scalability:** The proposed system uses a LoRa-based network architecture that supports flexible sensor node deployment. It is especially suited for environments where direct point-to-point communication is infeasible. Nodes can be added or removed without hardware modification, enabling wide-area scalability in large or dispersed beekeeping sites. In addition to network flexibility, the system supports use cases such as ecological monitoring, emergency deployments, and distributed smart agriculture. Its modular design improves system resilience and enables long-term deployment across diverse environmental regions [6][7].

**Accessibility:** A mobile application, developed using the Flutter framework, provides accessibility for non-technical users. This application is connected to a cloud-based server that receives real-time sensor data via MQTT. The server processes incoming data and ensures secure delivery to the mobile interface. The interface supports real-time monitoring of hive parameters including temperature, humidity, light intensity, and GPS location. When sensor data exceeds predefined thresholds, the system sends a notification to the user's mobile device. Real-time alerts allow timely intervention and improve decision-making during emergencies. This functionality enhances operational safety and system usability under field conditions [8][9].

**Comprehensive Hive-Environment Analysis:** Conventional beehive monitoring systems rely solely on internal sensor data, which limits the accuracy of condition assessment. In contrast, the proposed system monitors both internal and external variables to support contextual understanding of hive conditions. This integration helps distinguish between transient anomalies (e.g., internal temperature drops) and persistent external threats (e.g.,

shading, climate events), which supports more reliable decision-making. For example, a drop in temperature combined with low light and static GPS may indicate external causes rather than internal colony collapse. While not designed as a full ecological analytics platform, the system provides foundational environmental indicators that can be extended for advanced analytics in climate-resilient apiculture and smart agriculture [10][11].

**Low-Cost Implementation:** To ensure cost-effectiveness, the system replaces high-cost GSM/LTE modules with energy-efficient LoRa transceivers and uses low-cost sensors such as DHT22, LDR, and GPS modules. MQTT-based communication supports real-time data transmission and avoids the need for extensive network infrastructure. The system's architecture supports deployment in low-resource environments, such as rural apiaries, educational settings, and pilot-scale studies. Its minimal hardware footprint and ease of maintenance promote scalability and enable broader adoption in smart agriculture and environmental monitoring [12][13]. The proposed system offers a cost-efficient, scalable, and ecologically aware framework for real-time beehive monitoring, optimized for both technical deployment and real-world applicability.

#### A. Data Parameters for Project

Conducted over a two-month period under constrained time and funding, this study employed DHT22 temperature-humidity sensors, LDR, and GPS modules (NEO-6M) to collect data. A total of five worker nodes were deployed. Inter-node communication and data aggregation were handled via LoRa message-based transmission, and the master node relayed data to a central server using MQTT. The server handled real-time visualization and alert generation.

The system is inherently scalable. Its modular architecture and data processing routines support the addition of nodes across diverse terrains, enabling broad monitoring for large-scale apiaries. The system continuously collects and evaluates four key environmental parameters—temperature, humidity, light intensity, and GPS location—each of which plays a vital role in maintaining colony health and detecting anomalies. Detailed descriptions and corresponding threshold ranges for each parameter are as follows:

**Temperature:** Brood chamber temperature typically ranges between 32 °C and 36 °C, essential for larval development. Deviations below 32 °C or above 38 °C can elevate brood mortality or cause behavioral disruptions. More severe deviations (e.g., <30 °C or >40 °C) may result in irreversible colony damage. The system flags values outside [32 °C, 38 °C] as cautionary, and readings beyond [30 °C, 40 °C] as critical.[14]

**Humidity:** Optimal brood humidity lies within 50%–70%. Exceeding 75% may promote mold growth, while dropping below 45% can impair larval development. The system issues caution alerts outside [45%, 75%], and critical alerts beyond [40%, 80%].[15]

**Light Intensity:** Bee foraging is light-dependent, with activity diminishing under 100 lx. Readings below this threshold may indicate abnormal nocturnal movement or potential hive disturbance.[16]

**GPS Location:** Integrated GPS modules facilitate spatial mapping and anomaly localization. For instance, simultaneous high temperatures reported across adjacent hives may signal localized heat sources or fire hazards. GPS tracking also enables geofencing for unauthorized node relocation.[17] The acquired sensor data is continuously analyzed in real time, enabling prompt detection of anomalies and facilitating responsive, data-driven hive management.

#### B. Hardware Specifications

Each beehive contains one or more worker nodes, placed both inside and around the hive. The following hardware components were selected based on criteria such as low power consumption, outdoor reliability, and communication compatibility:

**Microcontroller Module:** An ESP32-based development board featuring a dual-core processor, integrated Wi-Fi and Bluetooth capabilities, and an onboard LoRa transceiver. Its low-power architecture and integrated peripherals make it highly suitable for battery-operated, connected remote nodes.

**Temperature & Humidity Sensor:** A composite digital sensor that integrates a capacitive humidity sensor and a thermistor. It outputs a calibrated digital signal on a single data line, eliminating the need for external analog-to-digital conversion and simplifying its integration with the microcontroller.

**Light Sensor:** LDR (Light Dependent Resistor), A variable resistor whose resistance decreases as the intensity of incident light increases. In this system, it is configured within a voltage divider circuit. The resulting analog voltage, which is directly proportional to the ambient light level, is measured by the microcontroller's ADC (Analog-to-Digital Converter) to determine light intensity.

**GPS Module:** NEO-6M GPS, A standalone GPS receiver module that tracks multiple satellites to provide precise location and timing data. It communicates with the microcontroller via a UART serial interface, transmitting standardized NMEA (National Marine Electronics Association) sentences that contain latitude, longitude, and other navigational information.

**LoRa Antenna:** A dipole antenna designed to operate efficiently in the sub-GHz ISM bands (e.g., 433/868/915 MHz) utilized by the LoRa module. Connected via an SMA connector, it is essential for achieving the long-range, low-power communication capabilities that define the LoRa protocol. The sensor data collected from these modules are formatted as JSON and transmitted to the master node via LoRa protocol. Full technical specifications such as

activation delay, operating voltage, and power consumption are summarized in Table I.

**Table I.** Technical specifications of the sensors used in the monitoring system

Sensor Type	Activation Time	Standby Power	Active Power	Operating Voltage
Temperature & Humidity (DHT22)	~ 2 sec	40–50 $\mu$ A	1–1.5 mA	3.3–6 V
Light Intensity (LDR)	Instant	<1 $\mu$ A	<0.5 mA	3.3–5 V
GPS Location (NEO-6M)	Several Seconds (hot start)	~ 6 mA (standby mode)	Up to 70 mA (active)	3.3–5.5 V

#### IV. SYSTEM ARCHITECTURE AND ROLE OF EACH COMPONENT

The proposed system architecture comprises five primary components: worker nodes for environmental sensing, a master node for data aggregation, an MQTT server for message brokering, a cloud server with a database for long-term storage and analysis, and a mobile application for real-time visualization and alerts.

##### A. System Architecture

The overall data flow begins with sensor readings at the worker nodes, which transmit the data to the master node via LoRa. The master node relays this data to the MQTT server over Wi-Fi, and the MQTT server forwards it to a cloud-based database. A mobile application subscribes to the MQTT topics to receive real-time updates and present them to end users. The system architecture is shown in Fig. 1.

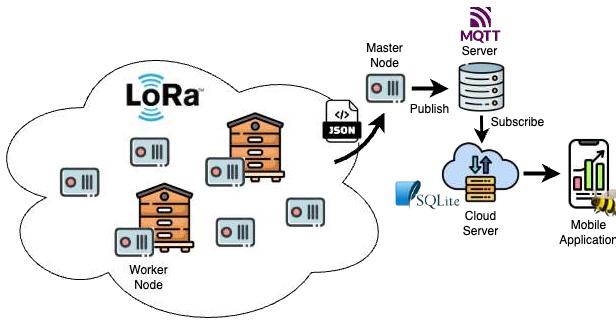


Fig. 1. Overall system architecture of the LoRa and MQTT-based beehive monitoring system.

##### B. Role of Each Component

**Worker Nodes:** Each beehive contains one internal worker node, installed within the brood chamber, tasked with monitoring temperature and humidity. Additionally, external nodes are deployed in the surrounding environment to capture light intensity and ambient temperature. For experimental validation, two worker nodes were deployed per hive: one internal node and one external node spaced at 1–2 meters from the hive walls. Each worker node is built around the Heltec Wi-Fi LoRa 32 V3 module, which integrates an ESP32 dual-core microcontroller, an SX1276 LoRa transceiver, and a 0.96-inch OLED display. This module serves as the core processing and communication unit. A 915 MHz SMA-connected dipole antenna is attached to each transceiver.

Although LoRa supports ranges up to 10 km in ideal line-of-sight conditions, our tests showed consistent connectivity up to 150 m in partially obstructed outdoor settings.

The physical implementation of the system, including the placement of worker nodes and hardware modules, is shown in Fig. 2. This prototype demonstrates the compact integration of sensors and LoRa-based communication components suitable for deployment around and inside beehives.

The detailed hardware configuration and data flow for each worker node are depicted in Fig. 3. This block diagram shows how sensor outputs are interfaced with the LoRa ESP32 microcontroller. The microcontroller is powered by a 3.7V Li-Po battery and integrates data from all sensors, packaging the output as a JSON object that contains environmental conditions and spatial-temporal metadata.

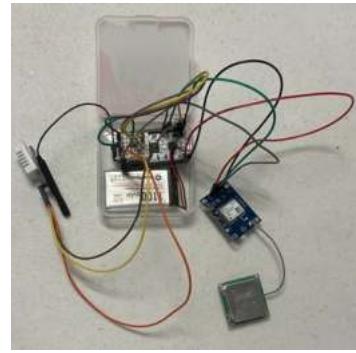


Fig. 2. Physical implementation of the proposed beehive monitoring prototype.

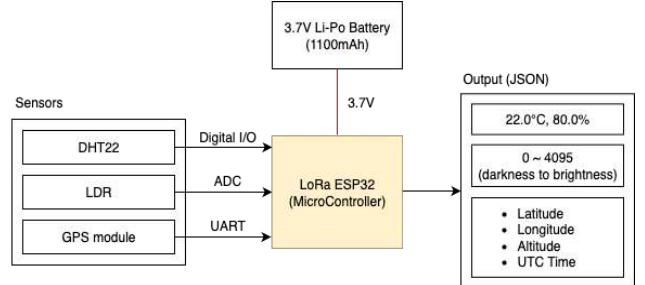


Fig. 3. Block diagram of the worker node.

The overall operation of the node follows a multi-stage process, starting with initialization, followed by periodic data collection and wireless transmission. During the initialization stage, the node configures the LoRa module, GPS setting and sets radio parameters. Setting the channel and sync word is essential to avoid collisions between nodes. In the main loop, the node collects sensor data every 3 minutes and formats it into a JSON packet. In the transmission stage, the JSON packet is sent to the master node via LoRa. The logical operation of each worker node, including sensor polling, JSON formatting, and LoRa transmission retries, is illustrated in Fig. 4. This block diagram provides a high-level abstraction

of the sensing-to-transmission cycle used by each node.

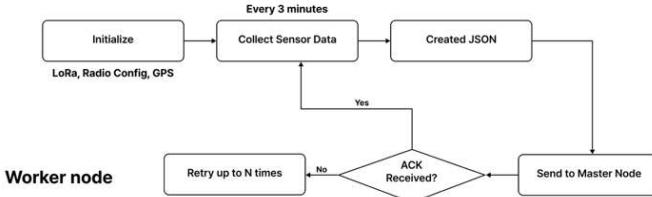


Fig. 4. Operation flow of the worker node: periodic sensing, JSON formatting, and LoRa transmission with retry logic.

**Master Node:** The master node employs the Wi-Fi LoRa 32 (V3) module and an external LoRa antenna to receive uplink data from surrounding worker nodes. Based on field experiments conducted in open field conditions, stable LoRa communication was maintained at distances up to approximately 110 meters, ensuring reliable long-range data collection in outdoor environments.

Functionally, the master node acts as a gateway between the LoRa and MQTT protocols. Upon receiving sensor data formatted in JSON from worker nodes, it synchronizes with a Network Time Protocol (NTP) server to append a precise UTC-based timestamp to each payload, thereby ensuring temporal consistency across all sensor readings. The JSON object contains multiple fields including temperature (in °C), humidity (in %), light intensity (as a percentage), GPS-based latitude and longitude, altitude, UTC time, and the corresponding node identifier. This enriched packet is then prepared for transmission to the cloud infrastructure.

The master node's operation is structured into three main stages: initialization, data reception, and MQTT publishing. Compared to worker nodes, the initialization stage incorporates additional setup routines such as Wi-Fi connection, MQTT client configuration, and NTP synchronization. During the data reception stage, incoming JSON packets are continuously monitored via the LoRa interface. Each packet is parsed and enriched with time metadata retrieved from the Wi-Fi-synchronized NTP server. In the transmission stage, the final JSON data is published to the MQTT broker under a predefined topic, allowing downstream services to store, visualize, or trigger alerts. The full operational workflow of the master node—including initialization, reception, timestamping, and MQTT publishing—is illustrated in Fig. 5.

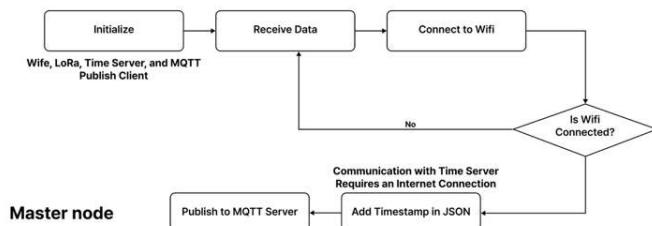


Fig. 5. Operational flow of the master node: data reception, timestamping, and MQTT transmission.

**MQTT Server:** The MQTT server handles routing of sensor data to subscribing clients using the publish/subscribe

paradigm. According to [18], MQTT shows 25% lower packet loss and 30% higher throughput in short-range communication scenarios (less than 5 km) when compared with LoRaWAN, making it suitable for small-scale distributed systems such as beehive monitoring. Since our deployment falls within this range, MQTT offers a more efficient and reliable communication solution for the system architecture.

**Cloud Server with Database:** The cloud server acts as the central data processing and storage unit within the system architecture. It continuously subscribes to relevant MQTT topics via the broker and stores the received sensor data in a SQLite relational database for long-term persistence and analysis. Sensor data is stored in the SQLite database for a minimum of 30 days, with periodic batch backups uploaded to encrypted cloud storage to ensure data reliability and fault tolerance.

RESTful APIs are implemented to facilitate secure communication between the server and the mobile application. The APIs support authentication using access tokens and implement a request rate-limiting policy of 60 calls per minute per client to ensure server stability. Furthermore, the APIs enable clients to query real-time sensor values, historical trend data, and anomaly reports filtered by location or node ID, thus supporting scalable application-layer analytics.

**Mobile Application:** The mobile application provides an intuitive user interface designed to assist beekeepers in monitoring and responding to environmental changes efficiently. The application was developed using Flutter, a cross-platform framework, allowing for deployment on both Android and iOS devices from a single codebase. The application offers three primary functions:

A. **Map Screen:** This screen displays the GPS locations of all deployed nodes using map-based visualization. Each marker includes dynamic pop-up information showing the node's real-time environmental readings such as temperature, humidity, and light intensity. The map is rendered using OpenStreetMap tiles and supports pan and zoom functionality. To enhance situational awareness, node markers are color-coded: green indicates normal status, red indicates a triggered alert, and yellow represents intermittent connectivity. This feature helps users identify the spatial distribution of sensor nodes and detect region-specific anomalies immediately.

B. **Data Visualization Screen:** Sensor data collected from the cloud server is visualized in the form of line graphs. Users can toggle between different time scales—daily, weekly, or monthly views—to observe patterns, identify fluctuations, and monitor trends that may impact hive health. The graphs update in near real-time and are color-coded for clarity.

C. **Emergency Alert System:** The application includes a built-in alert mechanism developed using Firebase Cloud Messaging (FCM) platform. Push notifications are triggered automatically when critical threshold values are detected. For instance, when the temperature exceeds 38°C or falls below 30°C in the brood chamber, or when light intensity exceeds 90% during nighttime, the system

issues a warning alert. These thresholds are based on field guidelines in [11].

To evaluate the system in real-world conditions, two types of outdoor environments were tested: a general building exterior and an actual bee yard.

#### A. Experiment at Building Exterior

**Objective:** The Objective of this experiment was to evaluate the reliability of sensor data collection and the performance of LoRa-based wireless communication. As LoRa performance can be affected by environmental factors such as obstacles, humidity, and temperature, this experiment was designed to establish baseline metrics in a semi-open environment. Specifically, the experiment aimed to confirm whether the worker nodes could consistently collect and transmit temperature, humidity, light, and GPS location data with minimal packet loss. In addition, this baseline test served to validate the system integration before deployment in the bee yard.

**Environment Description:** The test was conducted in the exterior area of a university building – specifically, in front of the K-SW building at Purdue University, Indiana, USA. The experiment took place on July 31, 2025, at approximately 2:00 PM on a sunny day with a temperature of around 28°C and high humidity, simulating typical outdoor conditions. The location was selected due to its semi-open structure, which included partial shading from trees and nearby buildings. All nodes-maintained line-of-sight (LOS) communication with the master node, but the presence of physical obstructions such as tree branches and benches introduced realistic transmission challenges.

**Node Configuration:** Three worker nodes were deployed, each equipped with environmental sensors. The nodes were positioned approximately 3 meters away from the master node, which served as the gateway device. The master node was placed inside the K-SW building to replicate a likely field deployment scenario in which the gateway device and Wi-Fi access point would be housed within a container or control shed adjacent to the beekeeping area. This configuration also allowed for a stable Wi-Fi connection to the MQTT server, mimicking real-world network constraints expected in remote monitoring environments.

**Data Collection:** Each node transmitted environmental data every 3 minutes, resulting in 30 transmission cycles over the 90-minute duration of the experiment. All three nodes consistently collected temperature, humidity, light intensity, and GPS location data without interruption. The collected data was formatted into structured JSON packets that included a timestamp, sensor readings, and spatial metadata.

**Results:** Lora-based wireless communication remained reliable over a distance of 10 meters in an open outdoor environment with minimal obstacles. Sensor data collected from all deployed nodes was consistent. The MQTT server and cloud server processed the incoming data without error. The mobile application successfully visualized the sensor readings and location information on the user interface.

To evaluate user interaction and real-time interpretation, a mobile application was developed. As shown in Fig. 6, the

application displays sensor values from each node, provides trend graphs for temperature, humidity, and sunlight levels, and issues emergency alerts when predefined thresholds are exceeded.



Fig. 6. Mobile application interface showing sensor data, trend graphs, GPS location of nodes and emergency alert.

#### B. Field Deployment in a Real Bee Yard

**Objective:** The objective of this experiment was to evaluate the system's performance under operational beekeeping conditions. Unlike controlled test environments, real bee yards introduce additional challenges such as hive structures that attenuate wireless signals and environmental noise. This experiment aimed to verify whether the system could reliably collect and transmit environmental data when nodes were placed both inside and around active beehives.

**Environment Description:** The experiment was conducted at an actual bee yard located on a farm near South 525 West, Indiana, USA. The test took place on August 4, 2025, at approximately 2:00 PM, during a moderately cool and humid afternoon. This environment was selected to simulate real-world deployment conditions, including physical obstructions such as hive boxes, vegetation, and varying terrain, as well as potential sources of wireless interference.

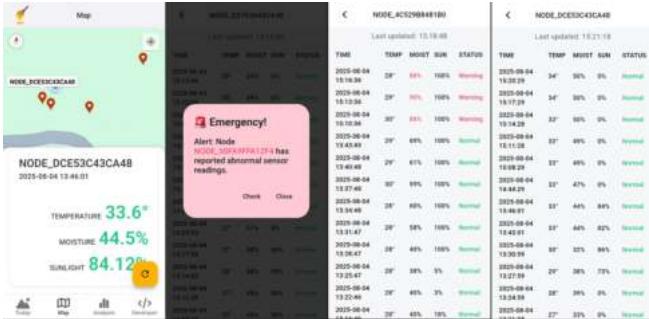
**Node Configuration:** A total of four worker nodes were deployed to enable direct comparison between internal and external hive environments. Two of the nodes were placed inside separate beehives to monitor internal environment. The remaining two nodes were installed just outside each corresponding hive to capture ambient environmental conditions. The master node was positioned approximately 30 meters away from the hives, inside a camper van located near the bee yard. This placement reflects a realistic field deployment scenario in which the gateway and Wi-Fi infrastructure are installed within a shelter or building commonly used by beekeepers.

**Data Collection:** During the experiment, all nodes transmitted environmental data at 3-minute intervals. Over a 1-hour testing period, each node generated approximately 20 data packets, all of which were successfully received by the master node. The internal nodes consistently recorded temperature and humidity values within the expected range for active hives. As shown in Fig. 7, the internal nodes recorded 0% light intensity, confirming that the light sensors functioned correctly in the absence of ambient light. This result validated the sensor's responsiveness to varying lighting conditions and demonstrated its ability to distinguish between indoor and outdoor placements. GPS location data from all four nodes

was received with coordinate variations within  $\pm 2$  meters, sufficient for verifying node placement. As illustrated in Fig. 7, the mobile application successfully visualized each node's location and sensor readings in real time. The interface also displayed dynamic alerts when sensor values exceeded predefined thresholds, further demonstrating the system's capability to support responsive and location-aware monitoring.

All nodes transmitted data every 3 minutes during the experiment. The internal nodes successfully collected temperature and humidity data from inside the hives. As shown in Fig. 7 the node placed inside the beehive recorded 0% light intensity, indicating that the light sensor functioned correctly. GPS data from all nodes was received without error. As shown in Fig. 7, the mobile application successfully displayed GPS positions, internal sensor readings, and real-time alerts for each deployed node.

**Results:** The experiment yielded the following results: LoRa communication remained stable even when nodes were placed inside wooden hive structures. Sensor data from both inside and outside the hives was collected accurately and consistently. The MQTT server and cloud server processed all incoming data reliably. The mobile application displayed hive-specific environmental data, confirming the correct integration of GPS-tagged measurements.



wireless communication was maintained even with sensors placed inside hive structures, and multi-parameter data—including temperature, humidity, light intensity, and geolocation—were successfully collected and analyzed. These results support the system's applicability for decentralized beekeeping operations, particularly in rural or low-infrastructure settings.

Future development will include hardware miniaturization, integration of advanced environmental and biological sensors, and implementation of adaptive routing protocols to improve network resilience. In addition, AI-based anomaly detection and long-term data logging will be incorporated to enable predictive analytics and autonomous decision-making. These enhancements aim to evolve the system into a scalable, intelligent platform for sustainable and data-driven apiculture.

#### ACKNOWLEDGMENT

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute of Information & communications Technology Planning & Evaluation) in 2025(2021-0-01393, 2022-0-01067, 2024-0-00064) and Purdue University. Sincere appreciation is extended to Professors Eric T. Matson and Anthony H. Smith at Purdue University for their valuable guidance on this work.

#### REFERENCES

- [1] P. Kamboj, G. Kaur, and G. Gupta, "Understanding the Impact of Heat Stress on Honeybee Populations: Insights into Thermoregulation, Colony Dynamics, and Environmental Interactions," \*Uttar Pradesh Journal of Zoology\*, vol. 45, no. 12, pp. 1–13, 2024. doi: 10.56557/upjzoz/2024/v45i124098.
- [2] A. Y. Kim, A. Velazquez, B. Saavedra, B. Smarr, and J. C. Nieh, "Exposure to constant artificial light alters honey bee sleep rhythms and disrupts sleep," \*Scientific Reports\*, vol. 14, no. 25865, pp. 1–10, 2024. doi: 10.1038/s41598-024-73378-9..
- [3] G. E. Leach and R. J. Paxton, "Exploring the external environmental drivers of honey bee foraging activity: A review of current knowledge and future directions," \*Biological Reviews\*, vol. 99, no. 3, pp. 1047–1063, 2024. doi: 10.1111/brv.13041
- [4] A. Nayyar, A. K. Singh, M. Sharma, and D. K. Jain, "HiveLink – IoT based smart bee hive monitoring system," \*Int. J. Adv. Comput. Sci. Appl.\* (IJACSA), vol. 16, no. 1, pp. 254–260, 2025.
- [5] R. Rahmatullah, R. Alkahfi, M. H. Purnomo, and M. A. Nugroho, "Multinode LoRa-MQTT design architecture and performance analysis for dual protocol network IoT," \*Int. J. Adv. Comput. Sci. Appl.\* (IJACSA), vol. 16, no. 1, pp. 296–303, 2025.
- [6] A. Ahmad, et al., "SPARC-LoRa: A Scalable Platform for Adaptive Remote Control Using LoRa," arXiv:2401.13569, 2024.
- [7] T. Adame, et al., "Understanding the limits of LoRaWAN," arXiv:1705.05899, 2017.
- [8] J. Moreno, et al., "Smart Beehive Monitoring using DHT22 and LoRa with Thinger.io," SSRN, 2024.
- [9] D. Liyanage, et al., "IoT-based Smart Beehive Monitoring System," ResearchGate, 2024.
- [10] D. Bortolotti, et al., "Beemon: A low-cost and scalable IoT platform for bee monitoring," \*Comput. Electron. Agric.\* , vol. 190, 2021.
- [11] S. El-Zoghby, et al., "IBSMC: An IoT-Based Beehive Smart Monitoring and Controlling System," \*Inventions\*, vol. 4, no. 3, 2019.
- [12] E. Ntawuzumunsi, et al., "SBMaCS: Smart Beehive Monitoring and Control System Using LoRa," \*Sensors\*, vol. 21, no. 10, 2021.
- [13] HiveMQ, "LoRaWAN and MQTT Integration in Smart Agriculture," [Online]. Available: <https://www.hivemq.com/blog/lorawan-and-mqtt-integrations-for-iot-applications-design/>
- [14] T. Stabentheiner, H. Kovac, and A. Brodschneider, "Honeybee colony thermoregulation – regulatory mechanisms and contribution of individuals in dependence on age, location and thermal stress," \*PLoS One\*, vol. 5, no. 1, e8967, 2010. doi: 10.1371/journal.pone.0008967.
- [15] A. Human, S. W. Nicolson, and V. Dietemann, "Do honeybees, *Apis mellifera scutellata*, regulate humidity in their nest?," \*Naturwissenschaften\*, vol. 93, no. 8, pp. 397–401, 2006. doi: 10.1007/s00114-006-0117-y.
- [16] A. Y. Kim, A. Velazquez, B. Saavedra, B. Smarr, and J. C. Nieh, "Exposure to constant artificial light alters honey bee sleep rhythms and disrupts sleep," \*Scientific Reports\*, vol. 14, no. 25865, pp. 1–10, 2024. doi: 10.1038/s41598-024-73378-9.
- [17] G. E. Leach and R. J. Paxton, "Exploring the external environmental drivers of honey bee foraging activity: A review of current knowledge and future directions," \*Biological Reviews\*, vol. 99, no. 3, pp. 1047–1063, 2024. doi: 10.1111/brv.13041.
- [18] P. F. E. Prasetyo, Ubaidi, Holipah, M. A. Mahmud, and R. Paradina, "Comparing the performance of LoRaWAN and MQTT protocols for IoT sensor networks," \*Journal of Information and Technology\*, vol. 6, no. 2, pp. 221–228, Jul. 2024, doi: 10.60083/jidt.v6i2.565.
- [19] Espressif Systems, "ESP32-S3 Series Datasheet," Apr. 10, 2022. [Online]. Available: [https://www.espressif.com/sites/default/files/documentation/esp32-s3\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf) (accessed Aug. 6, 2025).
- [20] K. Dokic, M. Vujoovic, and B. Malenovic Nikolic, "Is ESP32 with Camera Ready for Machine Learning?" Sensors, vol. 20, no. 14, p. 3936, Jul. 2020.
- [21] D. Ivkovic, Z. Stamenkovic, and I. Milinkovic, "Comparative Analysis of Microcontrollers for IoT Edge Devices," Facta Universitatis, Series: Electronics and Energetics, vol. 36, no. 1, pp. 95–110, Mar. 2023.