International Journal of Computing and Digital Systems
2025, VOL. 17, NO.1, 1-16

DOI:http://dx.doi.org/10.12785/ijcds/1570999725

Development of LoRa Multipoint Network Integrated with MQTT-SN Protocol for Microclimate Data Logging in UB Forest

Heru Nurwarsito[1] **and Mohammad Ali Syaugi Alkaf**[2]

[1,2]*Dept. Informatics Engineering, University of Brawijaya, Malang, Indonesia.*

Received 19 February 2024, Revised 16 September 2024, Accepted 3 October 2024

Abstract: The UB (University of Brawijaya) Forest, located on the slopes of Mount Arjuno in Indonesia, is a significant educational and research area with rich agricultural lands and diverse plant species. Traditional methods of microclimate data collection in this area have relied on manual sensor inspection by local farmers. This research introduces a novel approach by integrating Internet of Things (IoT) technology, particularly employing long-range (LoRa) communication, to overcome the limitations of conventional WiFi networks in remote data access. The implementation uses ESP32 modules for data transmission and reception, focusing on establishing a LoRa network compatible with the Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN) protocol. This enhances data exchange efficiency and reliability. The system is engineered to transmit 11 distinct microclimate data parameters every 15 minutes from two nodes. Preliminary testing reveals a maximum transmission range of 300 meters. However, the data loss rate is significant, averaging 50.49%, which reduces to 15% at a distance of 100 meters. Signal strength is strongest at -94 dBm for 100 meters and -121 dBm for 300 meters. Sensor calibration results indicate high accuracy, with soil moisture achieving 94.72% sensor accuracy and 99.76% calibration accuracy. These results, while promising, fall short of the LoRa Alliance's expected performance metrics, which suggest effective operational distances of up to 2 km under optimal conditions. This research demonstrates the potential and challenges of integrating IoT and LoRa technology in agricultural and environmental monitoring. The findings underscore the need for further optimization to achieve the range and reliability required for effective remote monitoring in rural and forested environments. This research sets a foundation for future enhancements in sensor network design and deployment strategies, aiming to improve data accuracy and accessibility for agricultural and environmental research.

Keywords: Internet of Things (IoT), LoRa, MQTT-SN, ESP32, Microclimate.

1. Introduction

In the digital age, technology significantly increases efficiency and productivity in many fields [1], including agriculture, a key export sector for Indonesia [2][3][4][5]. Effective monitoring and management are crucial for ensuring high yields. However, Indonesian farmers often face challenges in checking agricultural conditions in remote or mountainous areas due to infrequent manual monitoring and unreliable internet connections [6]. Research shows that the reach of WiFi in the rural and hilly regions of Indonesia is limited. WiFi is a technology that allows devices to connect to the internet within a specific range. WiFi is significantly limited due to geographical and infrastructural challenges. WiFi which typically functions well in urban areas, struggles to provide coverage beyond a maximum range of about 5.5 kilometers in such challenging terrains. This limitation hampers the accessibility of Internet services in these regions, thus affecting communication, education, and various other services that depend on reliable internet connectivity [7], but only 100 meters [8] with standard methods such as bamboo poles. Against this backdrop, LoRa (Long Range) technology stands out as a promising alternative, offering the ability to communicate over long distances with minimal power use. LoRa has been shown to send data up to 2800 meters, even in cities, making it a suitable option for rural agricultural monitoring [9].

To fully leverage LoRa's capabilities for communication across multiple sensors or nodes, integrating the LoRa protocol with Message Queuing Telemetry Transport for Sensor Networks (MQTT-SN) can streamline message transmission. MQTT-SN, which stands for MQTT for Sensor Networks, is tailored for sensor networks that operate under strict power and bandwidth limitations. This protocol is advantageous because it consumes less bandwidth

E-mail address: heru@ub.ac.id, saugialkaf1@student.ub.ac.id

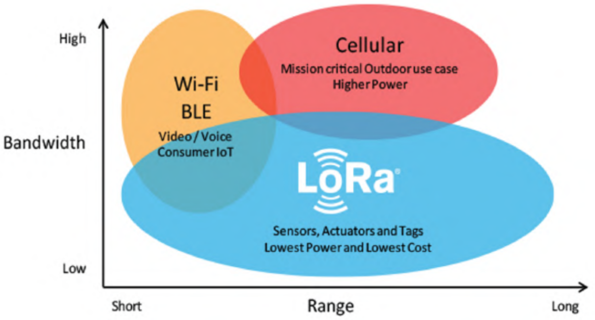

High
Bandwidth
Low
Short
Range
Long
Wi-Fi
BLE
Video/Voice
Consumer IoT
Cellular
Mission critical Outdoor use case
Higher Power
LoRa
Sensors, Actuators and Tags
Lowest Power and Lowest Cost

Figure 1. Comparison of LoRa Technology with WiFi and Cellular [12].

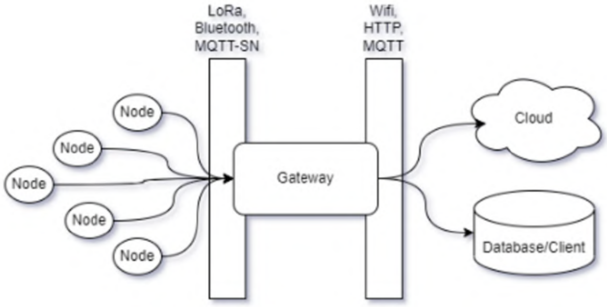


Figure 2. General Gateway Utilization

and power than its counterparts, MQTT and CoAP [10]. By merging LoRa networks with the MQTT-SN protocol, there is potential to significantly improve monitoring and management processes in agriculture, especially in areas where internet access is scarce. This approach promises to offer a viable and efficient solution for remote agricultural monitoring, making it easier to manage crops and predict yields in challenging environments.

2. Literature Review

A. LoRa (Long Range)

LoRa stands out as a wireless technology that prioritizes low power consumption and long-range communication, making it ideal for extending the battery life of devices while accommodating a wide network of connected devices. Its resilience and capacity are also noteworthy. LoRa uses the Chirp Spread Spectrum (CSS) for data transmission, which involves the use of varying radio frequencies to send data [7][9]. This method contributes to LoRa's effectiveness in maintaining communication over extensive distances, even in challenging environments, making it a preferred choice for various IoT applications.

LoRa technology is designed to use minimal bandwidth to enable long-range coverage, making it an efficient choice for wireless communication. In Indonesia, the specific frequencies for LoRa operation have been set following the public consultation decision on the RPM LPWA by ASIOTI, which allocated the LPWA frequency range to 920 – 923 MHz [11]. However, within the Indonesian LoRa community, such as TTN Indonesia, it is generally considered safe to use a frequency of around 921.4MHz for LoRa transmissions. This practice helps to avoid interference and ensures compliance with local regulations. Furthermore, in the context of research and practical applications in Indonesia, there is a tendency to focus on the use of LoRa at the physical layer rather than using the LoRaWAN protocol. This approach allows for direct control over transmission characteristics, which can be beneficial for custom applications and specific research needs, shown as in Figure 1.

In the selection of equipment for the research, various alternatives were considered to meet the specific requirements of the rural environment. While options like Zigbee and Wi-Fi were evaluated, LoRa technology was chosen for its long-range capabilities and low power consumption, making it ideal for remote agricultural monitoring. Zigbee, known for reliable short-range communication, did not offer the extensive coverage needed for remote applications. On the other hand, Wi-Fi, with higher data transfer rates, was limited by significant power consumption and shorter range, especially in challenging terrains like hilly or forested areas. The need for a low-power, long-range solution in the rural setting made LoRa the most suitable choice [13][14][15].

B. Internet of Things Gateway & End-node

The IoT Gateway plays a crucial role in the structure of IoT systems, acting as a bridge that connects IoT devices to the processing infrastructure [16][17]. It gathers data from multiple nodes, performs initial processing, and then relays this information to the cloud or a data center for deeper analysis [18]. On the other hand, IoT nodes are responsible for capturing environmental data via sensors and then communicating this data wirelessly with other systems [3] [19][20]. This setup allows for a seamless flow of information from the physical environment to the digital realm [21], where it can be analyzed and used to make informed decisions, enhancing efficiency and effectiveness across various applications [22][23]. The workings of the gateway within the Internet of Things framework can be seen, shown in Figure 2.

In this research, a specialized node was constructed through the integration of various components. The ESP32 module board Devkit V1 designed by Espressif Systems, was chosen for its strong features suitable for IoT applications, including a dual-core processor, Wi-Fi and Bluetooth capabilities, and multiple interfaces [20], shown in Figure 3. This board is used for both gateways and end nodes, details can be seen in Figure 13 and Figure 14.

Integrating the Internet with sensor networks in the research context is facilitated by utilizing a LoRa network in conjunction with the MQTT-SN protocol. This setup involves sensor nodes equipped with various environmental sensors that wirelessly transmit data via the LoRa network to a central gateway. The gateway, connected to the Internet through WiFi or another network interface, then forwards the sensor data to a cloud-based server for remote access and analysis. This configuration enables real-time monitor-

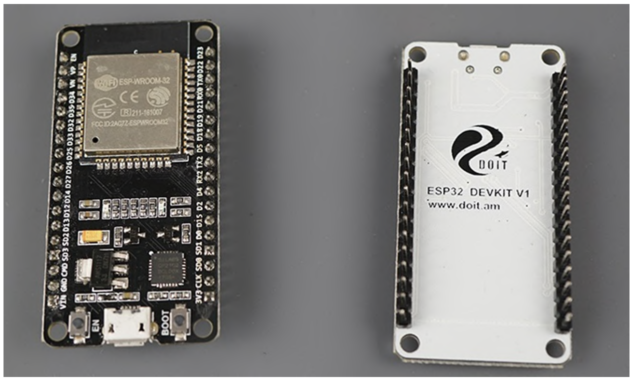


Figure 3. ESP32 DEVKIT V1 module board [20]

Figure 4. LoRa Hope-RFM9x wireless communication [27]

ing and management of microclimate data, even in remote or rural areas with limited traditional internet connectivity [24][25].

For wireless communication, the LoRa HopeRF RFM9x module was used, leveraging LoRa technology for effective long-distance data transmission [26], as shown in Figure 4.

The Capacitive Soil Moisture Sensor V1.2 was selected for its unique capacitive approach to measuring soil moisture [26], as shown in Figure 5.

While the DS18B20 was incorporated as a precise digital temperature sensor, its integration into the system involved meticulous calibration processes to ensure accurate and reliable temperature readings across various environmental conditions and applications [28] in Figure 6.

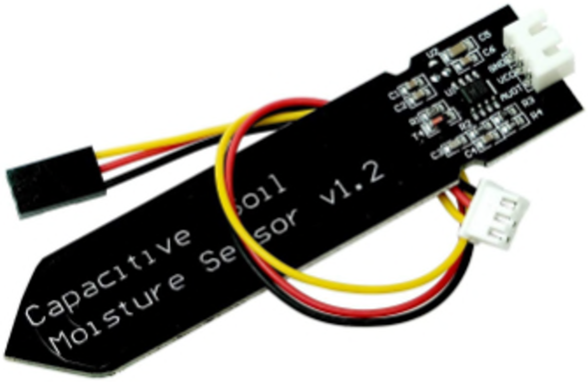


Figure 5. Capacitive Soil Moisture V1.2 [26]

Figure 6. DS18B20 digital temperature sensor [28]

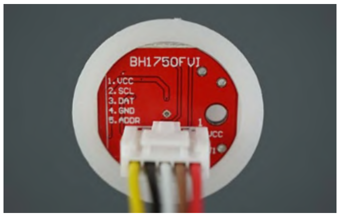


Figure 7. BH1750VI light intensity sensing [29]

Additionally, the BH1750VI was utilized for its high accuracy in light intensity sensing, enabling precise measurement and monitoring of ambient light levels essential for various applications, such as adaptive lighting systems and environmental monitoring solutions [29] in Figure 7.

DHT11 sensor was chosen for its simplicity and reliability in measuring temperature and humidity. These components were combined to create a comprehensive system capable of gathering and transmitting a wide range of environmental data, thereby facilitating enhanced IoT project functionality [30] shown in Figure 8.

C. MQTT-SN (Message Querying Telemetry Transport for Sensor Networks)

MQTT-SN, designed with sensor networks in mind, differs from the standard MQTT protocol in that it doesn't rely on the TCP/IP stack for communication but can use the UDP stack instead [31]. UDP, unlike TCP, does not require a connection setup before transmitting data and

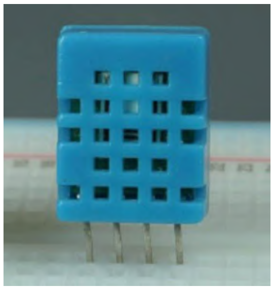


Figure 8. DHT11 temperature and humidity sensor [30]

does not guarantee ordered or error-free delivery of packets. This characteristic makes UDP faster and more efficient for applications prioritizing speed over reliability, such as real-time communications, streaming, and sensor networks [32]. UDP's compatibility with communication hardware, efficient transmission unit size, and high data transfer rates make it a preferred choice in scenarios where occasional data loss is acceptable [32].

Additionally, UDP supports transmission via broadcast and multicast, making it highly efficient for delivering data to multiple clients or devices on a network [33]. In various applications like real-time video services on the Internet, User Datagram Protocol (UDP) is predominantly used due to its connectionless nature and suitability for real-time streaming services, minimizing time overhead compared to TCP [34]. Furthermore, UDP is utilized in WebRTC for data transport, unlike other browser communication methods that rely on TCP, showcasing UDP's importance in specific communication technologies [35].

This adaptation makes it more suitable for environments where the traditional TCP/IP stack might be too heavy, such as in constrained devices or networks with limited bandwidth. One of the key distinctions of MQTT-SN is in its approach to topic identifiers. Unlike MQTT, which uses lengthy topic strings, MQTT-SN utilizes shorter Topic IDs, which can be 2-byte or even 1-byte integers, making it more efficient for scenarios with a large number of nodes. These shorter Topic IDs help in reducing the message size, further optimizing the protocol for low-bandwidth scenarios. In practice, nodes in an MQTT-SN setup communicate their messages to a dedicated MQTT-SN gateway [31]. This gateway then translates MQTT-SN topics into MQTT topics, allowing the messages to be seamlessly integrated into standard MQTT environments. This process enables devices using MQTT-SN to interact effectively with an MQTT server or broker [36], ensuring compatibility and facilitating message exchange within IoT ecosystems [37].

The establishment of a connection with the broker by MQTT-SN clients, also known as end nodes, necessitates communication through a dedicated MQTT-SN gateway. This setup ensures that messages from the end nodes can be properly formatted and transmitted to the MQTT broker, facilitating seamless integration into the broader MQTT ecosystem. Additionally, there's a component known as a gateway forwarder whose role is to facilitate two-way message exchange between end nodes when necessary [36], [38]. This forwarder acts as an intermediary, ensuring that messages can be relayed between end nodes, thereby enhancing the flexibility and connectivity options within the MQTT-SN network framework. This architecture is designed to support efficient communication in sensor networks and IoT applications where direct connection to a traditional MQTT broker may not be feasible or efficient due to network constraints or the nature of the devices involved. As depicted in Figure 9.

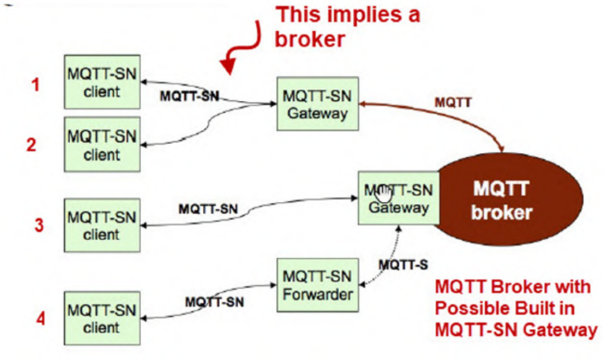


Figure 9. MQTT-SN Network Structure [31]

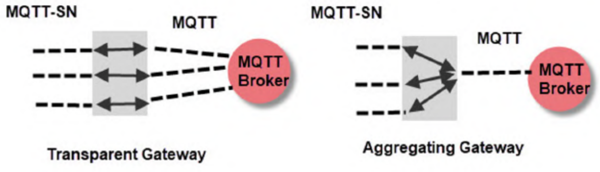


Figure 10. MQTT-SN Gateway Type [31]

In this research, the utilization of a forwarder is considered unnecessary as the primary objective is to collect sensor data from the end nodes. Instead, emphasis will be placed on employing a gateway functioning as an aggregator or topic combiner, as depicted in Figure 10, to facilitate the transmission of data to the broker. This approach streamlines the system's architecture by concentrating on the direct retrieval of sensor data without requiring intermediary message forwarding between end nodes.

The implementation of the MQTT-SN protocol within this research will be adapted to fit the LoRa network context. Key elements of MQTT-SN, such as the assignment of topic IDs and the adoption of a more streamlined message structure, will be integrated. These adaptations are aimed at optimizing message efficiency and reducing overhead, which is particularly important in the bandwidth-limited LoRa network environment.

Given that the system is designed to operate over a LoRa network, the implementation will not require the UDP protocol. This decision is based on the conceptual similarities between LoRa's communication model and the underlying principles of UDP [39], with a focus on minimizing complexity and leveraging the inherent long-range, low-power characteristics of LoRa technology. This approach allows for efficient, direct communication within the LoRa network, bypassing the need for the traditional network stack and thereby streamlining the process of data transmission from the sensor nodes to the central data processing hub.

D. LoRa Transmission Range

A couple of research conducted by Brawijaya's student [40][41] focused on setting up a LoRa network that included six end nodes and a single gateway. Their experiments were aimed at sending sensor data to the cloud and were carried

TABLE I. The International LoRa Developer Alliance reports

Spreading Factor (for UL at 125kHz)	Bit Rate	Range (depends on terrain)	Time on Air (for an 11-byte payload)
SF10	980 bps	8 km	371 ms
SF9	1760 bps	6 km	185 ms
SF8	3125 bps	4 km	103 ms
SF7	5470 bps	2 km	61 ms

out in the same area, specifically the UB forest. Since the results of their studies are currently confidential, In the paper, the positions of the end nodes were confirmed by the author with their advisor, and the previously utilized field site was personally checked by the author.

Following confirmation from their supervisor and an examination of the field site, data was gathered up to a range of 100 meters, with readings sent every 15 minutes. Observations indicated that beyond 100 meters, especially in wooded regions, previous studies reported transmission issues. The distance results from students's work exhibited variations compared to the data provided by the global LoRa developers' group.

In Table I, the International LoRa Developer Alliance reports that LoRa's minimum transmission distance is 2 km, which varies by location [12]. Their tests, mainly in urban and open areas, suggest different ranges. The minimum transmission distance in the UB Forest is expected to be less than 2 km due to obstacles like tall trees. This expectation is supported by previous studies conducted by Brawijaya's students and the advisor of the author. Augustin [9] found that LoRa's range reached 2300 meters with a Spreading Factor of 7 and 2800 meters in urban areas with a Spreading Factor of 12, using high-quality equipment like the KRDM-KL25Z board with a Semtech SX1276 MBED shield and a Cisco 910 gateway with a 6dBi Antenna.

3. Research Methodology

A. Research Location

This research examines the combination of LoRa and MQTT-SN protocols within an agricultural setting in the UB Forest, Karangploso, Indonesia. This area is characterized by its mountainous and sloping landscape, housing a small community of wooden and brick houses, with no tall buildings. The vicinity is surrounded by trees standing 10 to 30 meters tall. The region experiences regular rainfall in the afternoons and evenings during the rainy season. The research will collect data in a uniform format, capturing sensor readings, end-node IDs, and gateway RSSI values, sent directly from the end-points. Data gathering will occur from two points: a server that is directly linked to the gateway and clients that are connected to this server.

During the field data collection phase, the gateway will be set up in the UB Forest, equipped with an internet connection. Likely Figure 11, The end nodes will be strategically positioned at intervals starting from 1 meter, and then at 50 meters, 100 meters, 200 meters, 300 meters, 400 meters, and possibly beyond, contingent on the continued feasibility of data transmission. At each specified distance, each end node is tasked with transmitting 11 sets of data points.

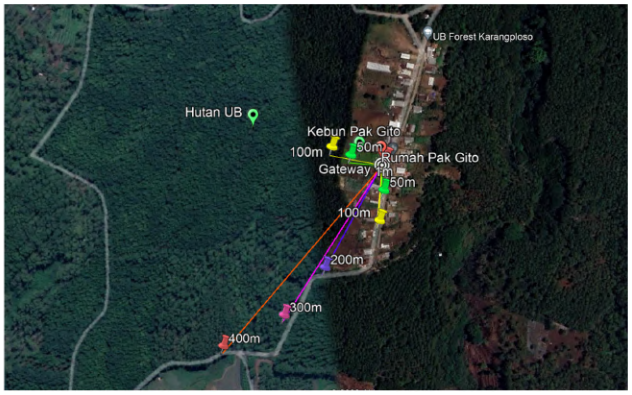


Figure 11. End-node Range Mapping

B. Requirements Engineering

In line with Table II and Table III, affordable sensors were chosen for microclimate monitoring due to budget constraints. For node 1 which is placed at the LC (Low Management Coffee) location, referring to the coffee-pine agroforestry area which is managed with minimal control or intervention, sensors for soil moisture, air humidity, soil temperature, and light intensity are used, while node 2 is placed at MC (Medium Management Coffee) locations are more active management, classified as low to medium, equipped with soil moisture and light intensity sensors. These sensors are not industrial quality to keep costs down.

The project uses a server, leased monthly or yearly, with adequate capabilities for data handling. This server functions as both a broker and a data repository for client access, suitable for small-scale research projects. Both the end nodes and gateways are powered by widely used microchips and share libraries like LoRa, PubSubClient, DHT, and DallasTemperature for operation. Node-RED, an IBM initiative, is installed on the server to provide an intuitive programming interface and automate data forwarding to clients.

C. Modeling & Implementing

The system's architecture is segmented into four key components based on their network functions, with a gateway serving as the connection point, shown in Figure 12. The components are the End-node, Gateway, Broker, and Client:

- End-node Component: This utilizes an ESP32 microcontroller for processing the data collected from environmental sensors, including a capacitive soil moisture sensor and a bh1750 light intensity sensor. The data, once gathered, is formatted into MQTT-SN messages, which are then encapsulated in LoRa

TABLE II. Hardware Requirements

Device Name	Description
MCU (Micro Chip Unit)	A microcomputer serves as the core processor to execute data retrieval/transmission commands for both end nodes and gateways using ESP32.
LoRa Module SX1276	Serves as a transceiver between sensor nodes and the gateway.
Sensors	Includes a Capacitive Soil Moisture Sensor for soil humidity detection and a BH1750 for light intensity measurement.
Server	Functions as an MQTT broker receiving message packets from the gateway and transmitting them to clients. Some servers are also utilized as clients.

TABLE III. Software Requirements

Software Name	Description
Operating System	Software platform utilized for program development and execution.
Libraries	Additional libraries facilitate programming on end-nodes and gateways, such as LoRa libraries for LoRa network communication and PubSubClient library for MQTT protocol implementation.
Node-Red	Application employed on the server to ease the creation of an MQTT broker and MQTT client within the same system.
Web Browser	Used for accessing the Thingspeak website on the client side.
Text Editor/IDE	The tool used for writing C language program code; in this research, IDEs like Arduino IDE are employed.

packets for transmission. The RFM9x module is used here for its LoRa communication capabilities, integrated with the MQTT-SN protocol.

- Gateway Component: Also powered by an ESP32 microcontroller, the gateway's main function is to receive the LoRa packets, decode the integrated MQTT-SN messages, and convert them into standard MQTT format. The RFM9x module aids in receiving data from the LoRa network. Simultaneously, the ESP32's WiFi capability is tasked with forwarding the MQTT-formatted data to the server where Node-RED is installed.
- Broker Component: On the server side, the Aedes MQTT Broker library is employed within Node-RED to set up the MQTT broker. This broker processes the data received from the gateway, organizes it by topics, and facilitates data access to clients through subscriptions.
- Client Component: Clients can be configured either on the same server as the broker or on external servers. For local access, a UI dashboard library on Node-RED displays the data on a webpage separate from the Node-RED editor. For external access, the setup uses Thingspeak as a client, leveraging the Thingspeak REST API for data input into the designated channel on Thingspeak.

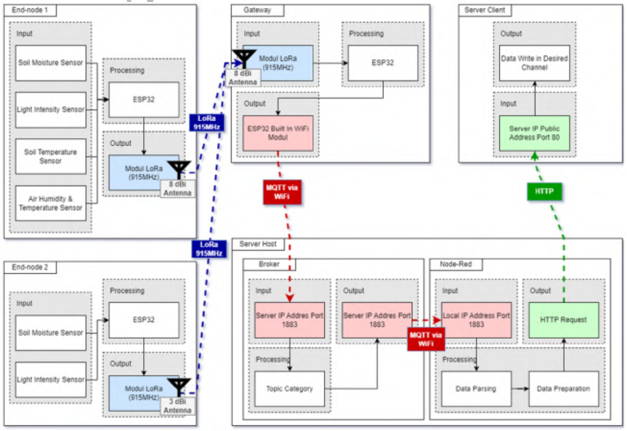


Figure 12. Overview of The System

This structure enables a seamless flow of data from the physical environment through sensor readings, across the network via LoRa and MQTT protocols, to end-users who can monitor and analyze the data through various interfaces.

The end node and the gateway are the two main components of the hardware architecture. As seen in Figure 13, the end node is outfitted with sensors that are interfaced with an ESP32 microprocessor to measure air humidity, temperature, light intensity, and soil wetness. The ESP32 processes the data collected from these sensors before formatting it into LoRa packets that use the MQTT-SN protocol for communication. Then, an RFM9x module is used to send this combined data, guaranteeing effective transmission over extended distances. Table IV provides a detailed explanation of the exact pin connections that enable communication between the RFM9x LoRa module, the ESP32, and the different sensors. This information helps replicate the system and comprehend its physical configuration.

On the other hand, the Gateway, as shown in Figure 14, is intended to accept data sent by the LoRa RFM95x module from the end node. This data is processed by an ESP32 microcontroller inside the Gateway after it is received. It is the responsibility of this microcontroller to reformat the received data into the MQTT protocol so that it may be sent again. After formatting, the data is transferred to a server's broker using the built-in WiFi module of the ESP32, guaranteeing smooth connectivity and data transfer to the network's subsequent phase. Table V provides comprehensive documentation of the pin connections that establish the

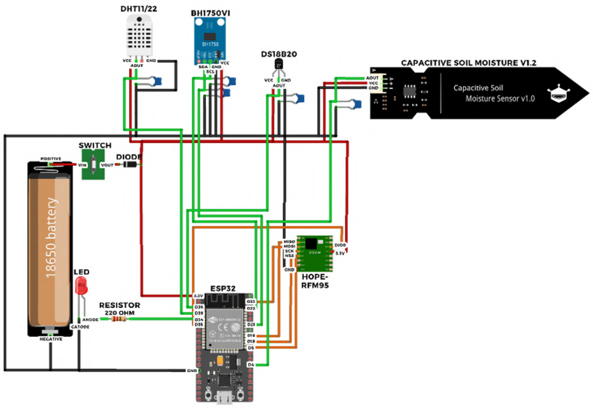


Figure 13. End-node Schematics

TABLE IV. End-Node Pin Configurations

Module Type	Module Pin	ESP32 Pin
Capacitive Soil Moisture Sensor V1.2	VCC, GND, Aout	VCC (3.3V), GND, D4 (GPIO)
Light Intensity Sensor BH1750VI	VCC, GND, SDATA, SCLK	VCC (3.3V), GND, D21 (SDA), D22 (SCL)
Soil Temperature Sensor DS18B20	VCC, GND, Aout	VCC (3.3V), GND, D25 (GPIO)
Air Humidity & Temperature Sensor DHT11	VCC, GND, Aout	VCC (3.3V), GND, D23 (GPIO)
LoRa Module (Hope-RFM95)	VCC, GND, NSS, RESET, SCK, MISO, MOSI, DIO0	VCC (3.3V), GND, D5 (SS), D14 (GPIO), D18 (SCK), D19 (MISO), D23 (MOSI), D35 (GPIO - Input Only)
LED	Anode, Cathode	D34 (GPIO), GND

interface between the ESP32 and the LoRa module within the Gateway. This information is crucial for configuring and setting up the hardware components of the Gateway.

The gateway comprises only ESP and LoRa components, acting as the central hub for data transmission. Enclosed in plastic casing and meeting specifications, it faces weather vulnerabilities, mitigated by protective measures like plastic covering in Figure 15(a). The end node includes ESP, LoRa, and four sensors, enclosed similarly in plastic and meeting the specifications in Figure 15(b). Vulnerable to weather, it also employs protective measures. Additionally, a PCB design enhances stability and performance for field deployment.

Figure 16 presents three distinct sections: (1) the LoRa Uplink Message Structure, (2) the Uplink Message Structure with MQTT-SN Integration, and (3) the MQTT-SN

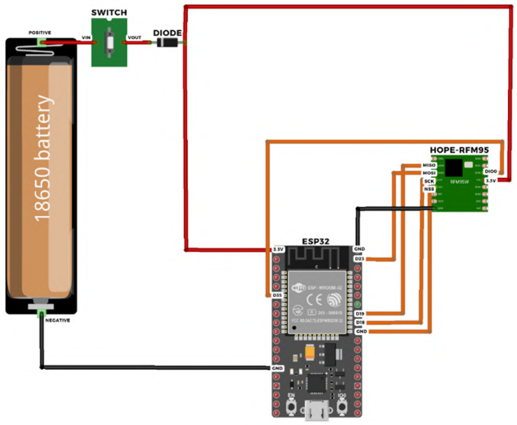


Figure 14. Gateway Schematics

TABLE V. Gateway Pin Configurations

Module Type	Module Pin	ESP32 Pin
LoRa Module (Hope-RFM95)	VCC, GND, NSS, RESET, SCK, MISO, MOSI, DIO0	VCC (3.3V), GND, D5 (SS), D14 (GPIO), D18 (SCK), D19 (MISO), D23 (MOSI), D2 (GPIO - Input Only)

Header Structure. The LoRa Uplink Message Structure (1) encompasses elements such as preamble, physical header (optional), physical header CRC (optional), physical payload, and CRC. With the integration of MQTT-SN, the Uplink Message Structure (2) expands to include additional components like MQTT-SN headers, while maintaining the foundational elements of the LoRa structure. This choice is driven by the need to efficiently organize incoming messages at the gateway, ensuring streamlined data sorting and processing without redundancy [31].

Moreover, the decision to not utilize the entire set

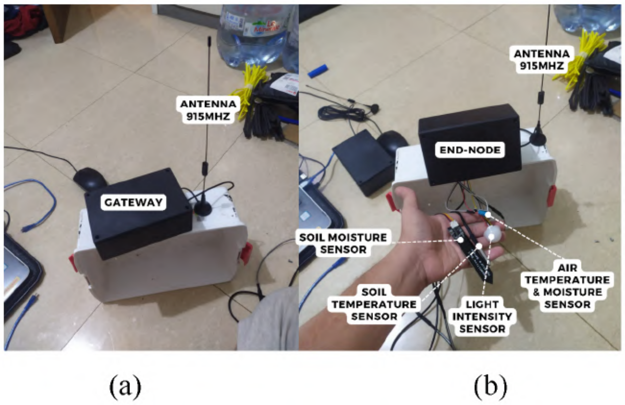


Figure 15. (a) Gateway (b) End-node

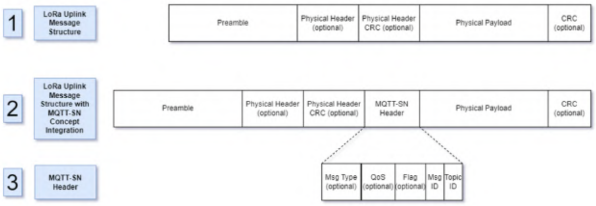

1
LoRa Uplink Message Structure
Preamble
Physical Header (optional)
Physical Header CRC (optional)
Physical Payload
CRC (optional)
2
LoRa Uplink Message Structure with MQTT-SN Concept Integration
Preamble
Physical Header (optional)
Physical Header CRC (optional)
MQTT-SN Header
Physical Payload
CRC (optional)
3
MQTT-SN Header
Msg Type (optional)
QoS (optional)
Flag (optional)
Msg ID
Topic ID

Figure 16. (1) LoRa Uplink Message Structure (2) Uplink Message Structure with MQTT-SN Integration (3) MQTT-SN Header Structure [31]

of MQTT-SN headers is influenced by the overlapping functionalities between MQTT-SN and LoRa protocols. The LoRa header adeptly manages crucial aspects such as message type, quality of service, and message flags, rendering the corresponding MQTT-SN headers unnecessary for this particular use case. This strategic approach of selectively incorporating headers optimizes the message structure to meet the unique requirements of the system, fostering a lean and efficient design of the communication protocol.

By carefully considering these factors and incorporating headers judiciously, the message structure is tailored to the specific needs of the system, enhancing its overall efficiency and effectiveness. This deliberate optimization ensures smooth data transmission, minimizes redundancy, and strengthens the reliability of the communication protocol in real-world applications.

Algorithm 1 provides a detailed pseudocode description of the End-node's programming of reading data and sending it to the gateway, meanwhile, Algorithm 2 shows an organized method of processing incoming packets in the gateway. There are multiple crucial steps in this process: First, data packets supplied across the LoRa network are received by the Gateway. It then starts the vital process of repackaging these data packets into the MQTT protocol format from their original LoRa format upon arrival. For the MQTT broker to handle and comprehend the data effectively, this modification is necessary. Lastly, the freshly formed MQTT data packets are sent to the broker via the ESP32 microcontroller of the Gateway's integrated WiFi module. Through this procedure, data may be seamlessly transferred from field-based physical sensors to a digital interface for monitoring, analysis, and action. It also creates a bridge between the LoRa network and the internet or local network.

A broker-functioning server within the system houses the Node-RED application, which is essential to the distribution and management of data. When data from the Gateway reaches the server, it is converted to JSON format. JSON standardization promotes accessibility and interoperability, making it simple for clients using different protocols to access and use the data. The research utilizes Thingspeak as a client platform, utilizing the HTTP protocol for data receipt. This highlights the adaptability of the system to accommodate several data transmission protocols.

Algorithm 1 End-node Pseudocode

```
Declaration of Libraries and Global Variables
Declaration of Pin Types, LoRa Network
Declaration of Data = 0
while [ do]
    Input Sensor Value
    Process: Pack sensor readings into MQTT-SN packet
    Process: Embed MQTT-SN packet into LoRa packet
    Output: Transmit data packet via LoRa module
    Increment Data
    if [ then ]Data=11
        Enter Deep Sleep for 60 seconds
        (Break WHILE)
    end if
end while
```

Algorithm 2 Gateway Pseudocode

```
Declaration of Libraries and Global Variables
Declaration of Pin Types, LoRa Network, WiFi Network
while [ do]
    Input Data Packet from LoRa network
    Process: Unpack the LoRa packet
    if [ then ]Topic Matches
        Process: Package payload into MQTT packet
        Output: Transmit data packet via WiFi module
    end if
end while
```

Crucial to the system's functioning, the server is set up with its public domain. This configuration guarantees that users can access data from any location with an internet connection, without being restricted by geography. As the MQTT broker, the server plays a fundamental role in the design of the system, going beyond simple data transit. Installing Node-RED on the server provides a flexible and intuitive environment for managing flow, processing data, and integrating with third-party services like Thingspeak. The system's whole workflow, from the point of data acquisition to the point of ultimate client distribution, is illustrated in Figure 17.

Physical sensors first gather the data, which is subsequently transmitted via the Gateway utilizing LoRa technology. The data is processed once it arrives at the Gateway and is then sent to the host server. The data is processed further once it is on the server, which serves as the MQTT

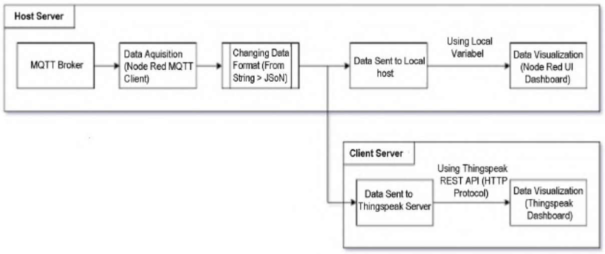

Host Server
MQTT Broker
Data Aquisition (Node Red MQTT Client)
Changing Data Format (From String > JSoN)
Data Sent to Local host
Using Local Variabel
Data Visualization (Node Red UI Dashboard)
Client Server
Data Sent to Thingspeak Server
Using Thingspeak REST API (HTTP Protocol)
Data Visualization (Thingspeak Dashboard)

Figure 17. Host Server's Flow

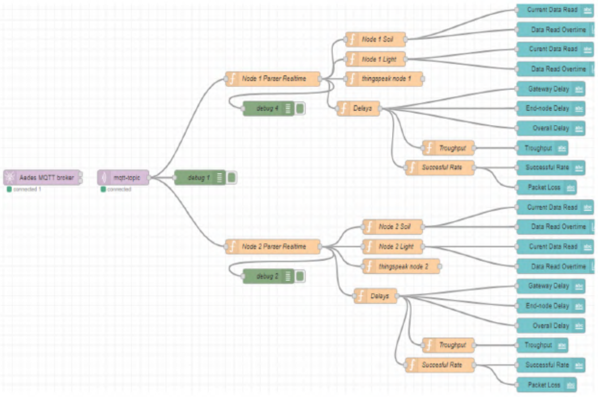
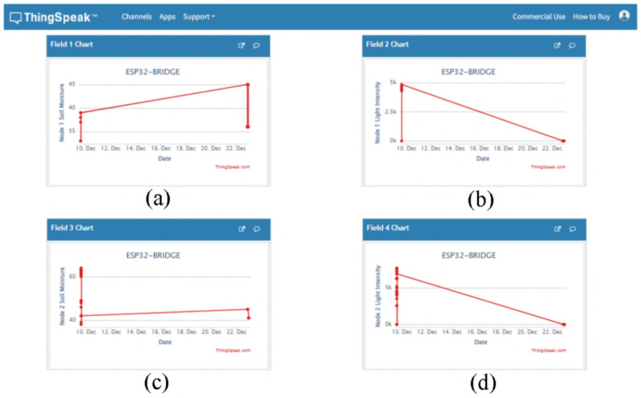


Figure 18. Node-red Flow Scheme

broker. Mostly, this involves converting the data into JSON format. This stage is essential for standardizing the data into a format that can be easily modified and recognized by a wide range of systems and applications. JSON is a lightweight format that is easily translated between different computer languages, making it perfect for Internet of Things applications where speed and efficiency are critical. Following the conversion to JSON, the data is prepared for client consumption. Because of the architecture's flexibility, this data can be accessed by a variety of devices; the only thing needed to change is the data transmission protocol according to the platform of the client. Thingspeak is the client platform used in this particular research; it was selected because of its interoperability with the HTTP protocol. This platform makes it possible to receive and visualize field data and provides an intuitive user interface for data analysis and decision-making.

This complete procedure, as shown in Figure 18, highlights how the system integrates hardware and software to enable a smooth information flow. It serves as an example of how an Internet of Things system may be made to be both resilient and adaptable, allowing for the collecting of environmental data in the real world and supporting many client platforms for data analysis and use. Within the broker server, the research utilizes its capabilities for both hosting the broker and executing data retrieval and processing tasks, streamlining the process for client reception [42]. The Node-RED application, a pivotal tool in this setup [43][44], facilitates the creation of a process flow diagram.

The Aedes MQTT broker node is essential to the beginning of this flow since it serves as the broker's starting point on the server and controls data flow inside the broker environment. To ensure that data coming in string format may be easily altered and delivered to several destinations, it is necessary to first capture the data and then transform it into JSON. This translation procedure is necessary to provide the flexible and effective transfer of values to Thingspeak and the dashboard server, which use the REST API or HTTP protocol for communication, respectively [45].



(a)

(b)

(c)

(d)

Figure 19. Data Display on Thingspeak (a) Node 1 Soil Moisture (b) Node 1 Light Intensity (c) Node 2 Soil Moisture (d) Node 2 Light Intensity

Figure 19 illustrates data display on ThingSpeak, showcasing readings from multiple nodes. (a) and (b) present soil moisture and light intensity data from Node 1 respectively, while (c) and (d) depict similar readings from Node 2. Each subplot provides a clear visualization of the respective sensor data, aiding in the analysis and monitoring of environmental conditions. To make it easier to monitor data trends in real time, this visualization technique necessitates the usage of Thingspeak's API for data writing and trend analysis [46]. The data from the graph can also be exported in CSV format for offline analysis or more in-depth review, providing a flexible method of handling and analyzing data.

This extensive configuration demonstrates how web platforms and software tools may be integrated to produce a smooth data flow from collection to visualization. To create a system that is effective and easy to use for monitoring and analyzing IoT data, the research skillfully used Node-RED to coordinate the data processing and Thingspeak to strategically display and analyze the data.

4. Result and Discussion

A. *Sensor Read Test*

Before sensor measurements, calibration is performed to determine the average accuracy [47][48]. Calibration involves comparing sensor measurements with manual measurements using a manual measuring device [49], as shown in Figure 20. AMF-035 is used for measuring air humidity, air temperature, and light intensity. Delta-T HH2 is used for measuring soil moisture, while Yieryi TPH01803 is utilized for measuring soil temperature.

In soil, air, and light classification, specific scenarios are created to determine the type of measurements obtained. For soil moisture scenarios shown in Figure 21, two soil samples (dry and moist) are collected from the research location. Moistening is achieved by adding water to the soil according to the prescribed amounts (no water for dry, half a container of water for moist). The same samples are used for soil temperature testing.

In Figure 22(a), the DHT sensor measures ambient hu-

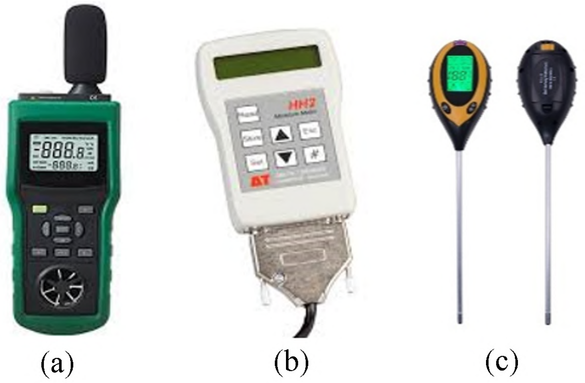


Figure 20. (a) AMF-035 (b) Delta-T HH2 (c) Yieryi TPH01803

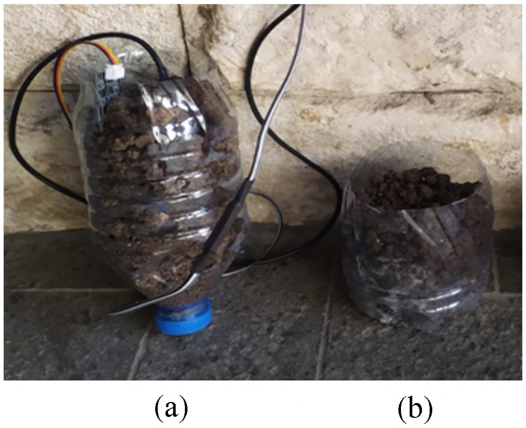


Figure 21. (a) Dry Soil (b) Moist Soil

midity, while in Figure 22(b), artificial humidity is created for testing. Tests on air humidity and air temperature were performed by the author using the DHT11 sensor in both normal and artificially humid conditions in an open room. Water droplets were sprayed toward the DHT sensor from a distance of 20-30cm to elevate air humidity. Sunlight was the only light source used for the light intensity test during the day and night. For soil, air, and light data classification, 11 data transmissions were performed in each experimental set with a 60-second interval, recording the highest value in each set.

The calibration of sensors took place at the UB Greenhouse location following the Equation 1 logic. Calibration was performed solely at this location, involving the collection of ten data points at 15-minute intervals. Additionally, manual measurements were taken using measuring instruments to serve as reference values for calibration.

Sensor accuracy refers to how close the sensor measurement results are to the actual value of the measured parameter. It indicates the sensor's ability to provide results that are correct or close to the true reference value. Calibration accuracy refers to the precision in establishing reference values for an instrument or sensor during the calibration process. It measures the extent to which an instrument or sensor conforms to a known and accepted standard or reference value.

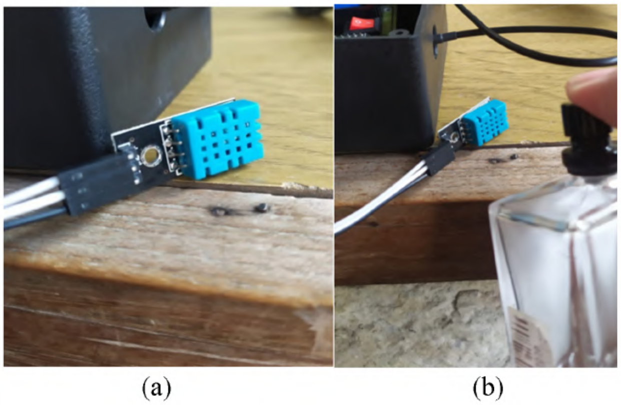


Figure 22. (a) DHT Sensor with Surrounding Humidity (b) DHT Sensor with Artificial Humidity

Testing Procedure:

- Before calibration, sensor readings were compared with manual microclimate measuring instruments for accuracy.
- Four sets of sensors from both nodes send data displayed via console or dashboard.
- Altering environmental conditions to test sensor responsiveness to changes.
- Recording data and comparing sensor readings with expected values.

From the calibration of each sensor, linear regression was applied to each measurement to determine the extent of the difference between the sensor measurements and the manual measurement tool, thereby assessing calibration accuracy [50]. To determine the average calibration accuracy and track the trend of the data changes, each sensor underwent ten calibration tests. A statistical technique for simulating the linear relationship between two or more variables is called linear regression. Linear regression is used in the context of sensor calibration to determine the relationship between sensor readings (x) and manual measurement readings (y).

$$y = mx + c \tag{1}$$

Where in the context of this research:

- (y) is the result of calibration or manual measurement reading.
- (m) is the slope of the regression line, indicating the magnitude of change in the expected manual measurement reading when the sensor reading changes.
- (x) is the sensor reading.
- (c) is the intercept or constant coefficient of the regression result.

Using linear regression, it can be understood how well

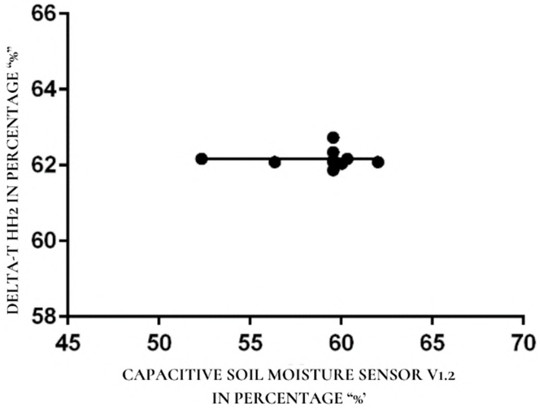


Figure 23. Soil Moisture Regression Graph

TABLE VI. Soil Moisture Calibration and Accuracy Result

Soil Moisture Sensor V1.2 (%)	Delta-T HH2 (%)	Calibration (%)	Sensor Accuracy (%)	Calibration Accuracy (%)
52.34	62.17	62,17	84,18	99,99
56.35	62.08	62,17	90,76	99,85
59.56	62.13	62,17	95,86	99,93
59.56	62.08	62,17	95,94	99,85
60.04	62.04	62,17	96,77	99,79
59.56	62.34	62,17	95,54	99,72
59.56	62.73	62,17	94,94	99,11
59.56	61.87	62,17	96,26	99,51
62.02	62.08	62,17	99,90	99,85
60.34	62.17	62,17	97,05	99,99
		Average Accuracy	**94,72%**	**99,76%**

the sensor readings can predict manual measurement readings [51][48], additionally, it is possible to ascertain whether the sensor measurements include any systematic bias or error. Information regarding the statistical significance of this link is also provided via regression analysis. Figure 23 to Figure 27 and Table VI to Table X will explain each sensor reading's linear regression analysis results.

As shown in Table XI, the average sensor accuracy of soil moisture is 94.72%, soil temperature 88.79%, air humidity 92.13.36%, air temperature 80.49%, and light intensity 89,85%. Regarding the average calibration accuracy, the highest percentage is 99.76% for Soil Moisture, followed by 97.26% for Air Temperature, 95.56% for Air Humidity, 94.28% for Soil Temperature, and 84.85% for Light Intensity. These findings indicate that the sensors have been calibrated properly using linear regression. The average accuracy following calibration shows a notable improvement, indicating that the calibration procedure is successful in raising the accuracy and linearity of sensor data, since it was linear, the sensor was proven to be efficient [52]. In addition, the linear regression equations show a

TABLE VII. Soil Temperature Calibration and Accuracy Result

DS18B20 (°C)	Yieryi TPH01803 (°C)	Calibration (°C)	Sensor Accuracy (%)	Calibration Accuracy (%)
27	31	30	87.09	96.77
27	30	30	90	100
27	31	30	87.09	96.77
27	31	30	87.09	96.77
27	26	30	96.15	84.62
26	29	29	89.65	100
26	31	29	83.87	93.55
26	31	29	83.87	93.55
27	31	30	87.09	96.77
26	25	29	96	84
		Average Accuracy	**88.79%**	**94.28%**

TABLE VIII. Air Humidity Calibration and Accuracy Result

DHT11 (%)	AMF-035 (%)	Calibration (%)	Sensor Accuracy (%)	Calibration Accuracy (%)
58.2	45.5	48.31	72.09	93.82
60.5	59.0	57.44	97.46	97.36
60.0	55.0	55.46	90.91	99.16
63.2	65.5	68.16	96.49	95.93
61.0	59.5	59.43	97.48	99.88
61.0	62.3	59.43	97.92	95.39
60.4	55.1	57.05	90.38	96.46
60.8	53.9	58.64	87.19	91.21
60.5	65.0	57.44	93.08	88.37
60.8	59.8	58.64	98.33	98.05
		Average Accuracy	**92,13%**	**95,56%**

TABLE IX. Air Temperature Calibration and Accuracy Result

DHT11 (°C)	AMF-035 (°C)	Calibration (°C)	Sensor Accuracy (%)	Calibration Accuracy (%)
20.0	26.8	27.09	74.63	98.89
21.0	27.9	27.80	75.27	99.65
21.0	28.2	27.80	74.47	98.59
25.0	28.9	30.62	86.51	94.05
25.0	30.0	30.62	83.33	97.93
25.0	29.9	30.62	83.61	97.59
26.0	30.5	31.32	85.25	97.29
25.0	31.8	30.62	78.62	96.29
25.0	32.9	30.62	75.99	93.07
30.0	34.4	34.142	87.21	99.25
		Average Accuracy	**80,49%**	**97,26%**

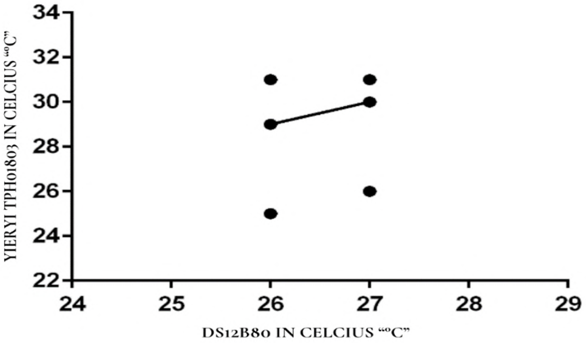


Figure 24. Soil Temperature Regression Graph

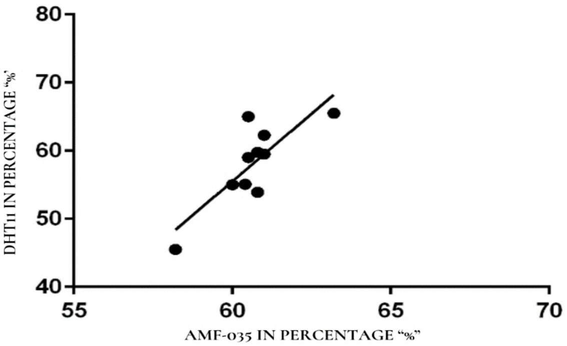


Figure 25. Air Moisture Regression Graph

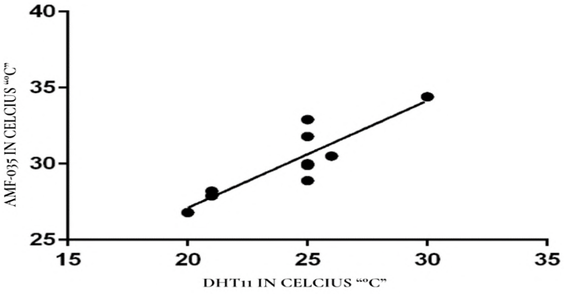


Figure 26. Air Temperature Regression Graph

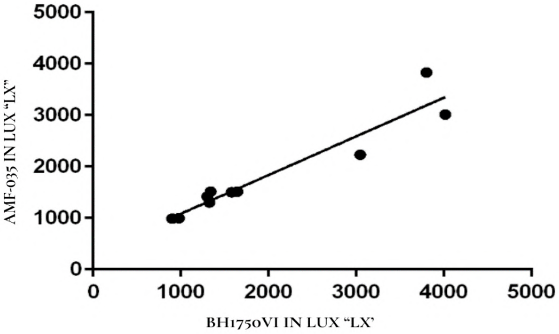
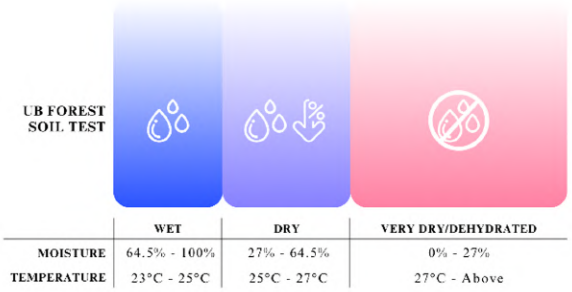


Figure 27. Light Intensity Regression Graph

TABLE X. Light Intensity Calibration and Accuracy Result

BH1750VI (lx)	AMF-035 (lx)	Calibration (lx)	Sensor Accuracy (%)	Calibration Accuracy (%)
3800	3830	3214.4	99.21	84.59
1642	1513	1469.9	92.14	89.52
3046	2230	2009.7	73.21	65.98
4015	3015	2600.8	75.09	64.77
980	1000	1083.7	97.96	89.42
900	990	1076.2	90	80.43
1325	1300	1309.6	98.11	98.84
1340	1513	1469.9	87.09	90.30
1300	1420	1399.9	90.77	92.31
1580	1500	1460.2	94.94	92.41
		Average Accuracy	**89.85%**	**84.85%**



UB FOREST SOIL TEST	WET	DRY	VERY DRY/DEHYDRATED
MOISTURE	64.5% - 100%	27% - 64.5%	0% - 27%
TEMPERATURE	23°C - 25°C	25°C - 27°C	27°C - Above

Figure 28. Soil Sensor Testing

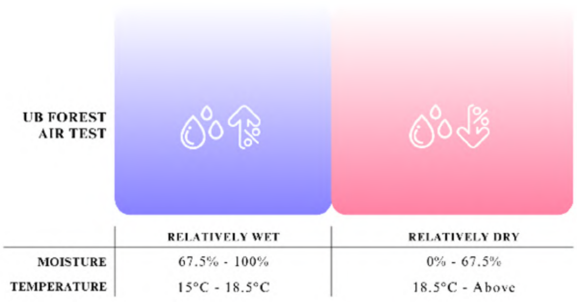
robust correlation between the measured values and the sensor readings, and the regression Equation agrees well with the features of each sensor.

Figure 28 to Figure 30, Regarding the microclimate readings in the UB Forest, a tentative conclusion can be made. Three categories of microclimates—soil, air, and light—are applied to the readings. Based on the gathered sample data, sensor reading thresholds linked to predicted outcomes have been assigned to each category. The soil temperature, soil humidity, and light intensity sensors, out of the four examined sensors, show acceptable linear detection capability with changing environmental circumstances and somewhat sensible data collecting for the sensor data for the UB Forest region. However, using high-accuracy analog climate-detecting methods requires recalibration. Because it takes longer to read data, the DHT sensor still has trouble producing consistently accurate results. As a result, the DHT22 sensor will be employed, and the circuitry's power input will be reassessed.

Integrated LoRa with MQTT-SN Performance Test Evaluating the performance of the LoRa network combined with the MQTT-SN protocol is the aim of this testing. The exact delay cannot be determined because the ESP32 only has a timer in its RTC. As an alternative, the number of packets that are successfully received in a single set of tests, as well as the successful rate, will be counted by measuring the last

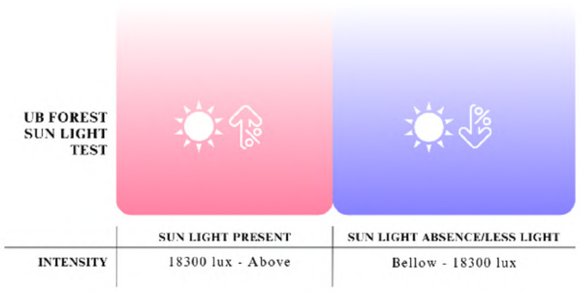
TABLE XI. Linear Regression Equation and Accuracy

Sensor	Regression Equation	Average Sensor Accuracy	Average Calibration Accuracy
Soil Moisture	y = -0.0002464x + 62.17	94.72%	99.76%
Soil Temperature	y = 1.000x + 3.000	88.79%	94.28%
Air Humidity	y = 3.971x - 182.8	92.13%	95.56%
Air Temperature	y = 0.7044x + 13.01	80.49%	97.26%
Light Intensity	y = 0.7529x + 330.8	89.85%	84.85%



UB FOREST AIR TEST	RELATIVELY WET	RELATIVELY DRY
MOISTURE	67.5% - 100%	0% - 67.5%
TEMPERATURE	15°C - 18.5°C	18.5°C - Above

Figure 29. Light Sensor Testing



UB FOREST SUN LIGHT TEST	SUN LIGHT PRESENT	SUN LIGHT ABSENCE/LESS LIGHT
INTENSITY	18300 lux - Above	Bellow - 18300 lux

Figure 30. Air Sensor Testing

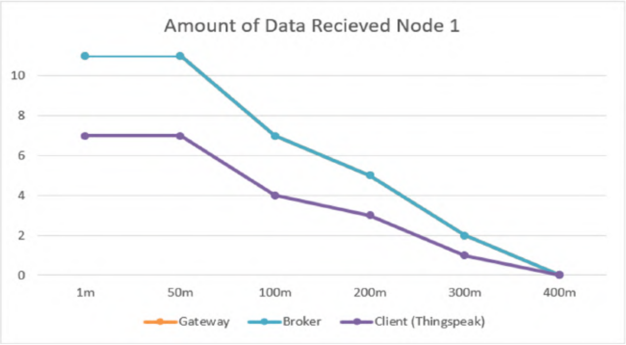


Figure 31. Graph of Sent Packets Testing Node 1

received signal strength indication (RSSI). This indication is delivered from the gateway to each end node and can be used to validate signal performance [3] and evaluate the number of packets successfully received in a single set of experiments or the successful rate. In this case, measurements will be taken at multiple sites to achieve different RSSI values and successful rates. Distances of 1, 50, 100, 200, 300, and 400 meters will be measured. One round of data transfers—totaling 11 data transmissions—will occur during each distance measurement. By placing the end node at different distances, performance discrepancies are hoped to be identified. It is significant to note that the transmission range is limited by the utilization of a 3.3V-based antenna.

Testing Procedure:

- Place the end node at the designated distance for measurement.
- Transmit data using the LoRa network integrated with the MQTT-SN protocol and then send it to the client using the HTTP protocol.
- Record the amount of data received at the gateway, broker, and client.
- Repeat the process with different transmission distances.

The results demonstrate that the data transmission performance is significantly affected by distance and obstacles. The details regarding data transmission per end-node. In Figure 31 and Figure 32, data transmission is successful from 1m to 50m, but packet loss begins to occur at 100m. The difference between Node 1 and Node 2 becomes apparent at 100m, where Node 1, using the antenna, sends more data and experiences fewer packet losses. Node 1 can also transmit data further than Node 2. It may not satisfy the LoRa official range capability as it has been tested in urban areas with a capability of 2800m [9]. But it surpasses the research before at the same place UB Forest which only reaches the range of 150m to 200m [40][41].

In Figure 33, signal strength is measured using RSSI, and it was significantly impacted by distance and obstacles. At 1m and 50m, where there are relatively few obstacles, high RSSI values are observed. However, at 100m, RSSI starts to decrease notably due to increased distance and obstacles like tall trees and buildings. While Node 1 maintains strong transmission capabilities at 300m, Node 2 cannot transmit data at that distance. Beyond 300m, data transmission becomes impossible. From the collected data, it is evident that distance and obstacles greatly impact signal strength and the amount of data sent in a specific transmission set. A relationship with packet loss was concluded, as depicted in Figure 34 and Figure 35.

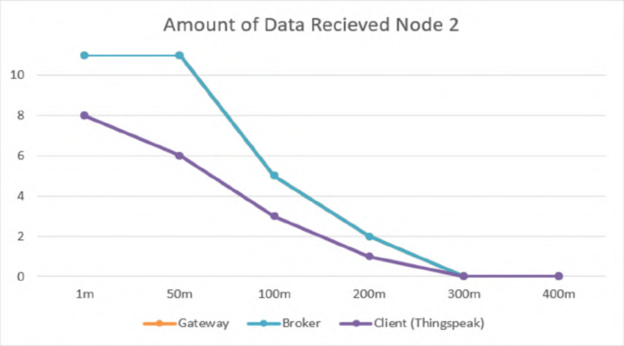


Figure 32. Graph of Sent Packets Testing Node 2

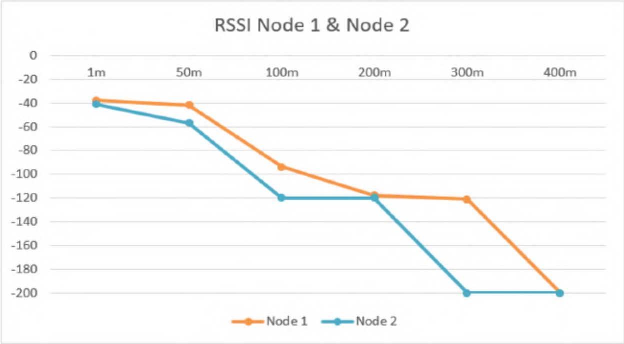


Figure 33. RSSI Testing Graph

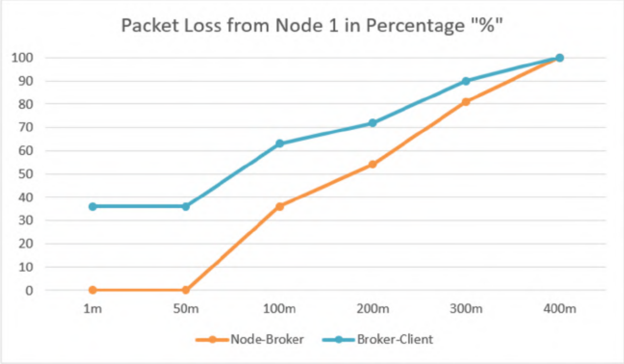


Figure 34. Graph of Packet Loss Node 1

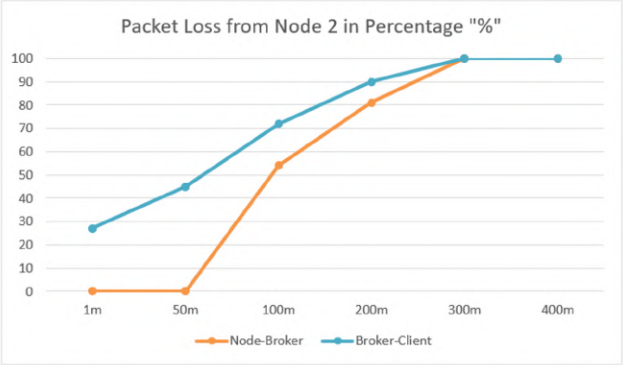


Figure 35. Graph of Packet Loss Node 2

The packet loss values were obtained by calculating received data divided by transmitted data, as in Equation 2.

$$PacketLoss = 100\% - \frac{Amount\ of\ Data\ Recieved}{Amount\ of\ Data\ Sent} x100\% \quad (2)$$

The average packet loss from Node 1 to the broker is 45.16%, and from Node 2 is 55.83%, resulting in an average packet loss between them of 50.49%. These figures are relatively high and unreliable. However, by reducing the measurement distance to 100m, the average packet loss for Node 1 becomes 12% and for Node 2 becomes 18%. Combining these averages yields an overall packet loss average of 15%, which remains considerably high and requires significant hardware and software configurations [53].

5. Conclusions and Future Work

The development of the microclimate data logging system has been successfully implemented. The sensor data acquisition process demonstrates successful calibration using linear regression, resulting in improved accuracy across various sensors. Specifically, soil moisture sensors achieve an average accuracy of 94.72% and a calibration accuracy of 99.76%. Soil temperature sensors reach an average accuracy of 88.79% and a calibration accuracy of 94.28%. Air humidity sensors show an average accuracy of 92.13% with a calibration accuracy of 95.56%, while air temperature sensors exhibit an average accuracy of 80.49% and a calibration accuracy of 97.26%. Light intensity sensors record an average accuracy of 89.85% and a calibration accuracy of 84.85%. The application of linear regression effectively corrects discrepancies between sensor readings and reference values, thereby enhancing the sensors' ability to detect microclimates within the UB Forest. The data transmission performance using the MQTT-SN protocol over LoRa is significantly affected by distance and obstacles. Optimal transmission is observed within a range of approximately 100 meters, showing an average packet loss of 15% and an RSSI value of around -94 dBm. However, beyond this range, such as at 300 meters, the average packet loss increases to 50.49%, with an RSSI value of approximately -121 dBm.

Based on the testing outcomes and literature review, several recommendations are proposed. Firstly, the research protocol's application should be extended beyond agriculture to include sectors such as logistics and maritime. This broader application could provide a more comprehensive understanding of the protocol's potential use cases. Furthermore, exploring the possibility of bidirectional data transmission could enable remote triggering capabilities, allowing functionalities such as relay activation and interval adjustments at the endpoints. This enhancement could significantly increase the versatility and functionality of the sensor network.

References

[1] C. Jin, A. Xu, Y. Zhu, and J. Li, “Technology growth in the digital age: Evidence from china,” *Technol Forecast Soc Change*, vol. 187, no. 18, pp. 122 221,.

[2] Z. Yao, H. Zhu, and W. Du, “Design and implementation of automated warehouse monitoring system based on the internet of things,” *Applied Mechanics and Materials*, vol. 547, no. 1, pp. 1099–1102,.

[3] Y. Noprianto, V. Firdaus, M. Mentari, I. Siradjuddin, and M. Kusuma, “The application of lora module and smart card for a large-scale area attendance monitoring system,” *International Journal of Computing and Digital Systems*, vol. 14, no. 1, pp. 815–826,.

[4] U. Heriqbaldi, A. Jayadi, A. Erlando, B. Samudro, W. Widodo, and M. Esquivias, “Survey data on organizational resources and capabilities, export marketing strategy, export competitiveness, and firm performance in exporting firms in indonesia,” *Data Brief*, vol. 48.

[5] Kemendag, “Total ekspor impor satu data perdagangan,” available:. [Online]. Available: https://satudata.kemendag.go.id/data-informasi/perdagangan-luar-negeri/ekspor-impor

[6] A. Cahyani, N. Nachrowi, D. Hartono, and D. Widyawati, “Between insufficiency and efficiency: Unraveling households’ electricity usage characteristics of urban and rural indonesia,” *Energy for Sustainable Development*, vol. 69, pp. 103–117,.

[7] H. Hariyanto, H. Santoso, and A. Widiawan, “Emergency broadband access network using low altitude platform.”

[8] T. Rosdiyani and N. Setiawan, “Pemasangan jaringan internet berbasis wireless fidelity (wifi) di kampung,” pp. 181–191,.

[9] A. Augustin, J. Yi, T. Clausen, and W. Townsley, “A study of lora: Long range low power networks for the internet of things,” *Sensors (Switzerland*, vol. 16, no. 9, pp. 1–18,.

[10] M. Marti, C. Garcia-Rubio, and C. Campo, “Performance evaluation of coap and mqtt-sn in an iot environment,” pp. 1–12,.

[11] D. Sallyna, U. K. Usman, and M. Murti, “Perencanaan jaringan long range (lora) pada frekuensi 920 mhz-923 mhz di kota bandung long range (lora) network planning with frequency 920 mhz-923 mhz in bandung city,” *e-Proceeding of Engineering*, vol. 7, no. 1, pp. 933–940, 2020.

[12] L. Alliance, “What are lora® and lorawan®?” *Semtech.com. Accessed*, online]. Available:. [Online]. Available: https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/

[13] B. Dash and J. Peng, “Zigbee wireless sensor networks: Performance study in an apartment-based indoor environment,” *Journal of Computer Networks and Communications*, vol. 2022, pp. 1–14,.

[14] S. Edirisinghe, A. Wijethunge, and C. Ranaweera, “Wi-fi 6-based home area network for demand response in smart grid,” *International Journal of Communication Systems*, vol. 37, no. 9.

[15] P. Visconti, R. Fazio, R. Velázquez, C. Del-Valle-Soto, and N. Giannoccaro, “Development of sensors-based agri-food traceability system remotely managed by a software platform for optimized farm management,” *Sensors*, vol. 20, no. 13, pp. 3632,.

[16] J. A, R. Chakravarthy, and M. L, “An experimental study of iot-based topologies on mqtt protocol for agriculture intrusion detection,” *Measurement: Sensors*, vol. 24, no. August, pp. 100 470,.

[17] P. Prakash, D. Kavitha, and P. Reddy, “Delay-aware relay node selection for cluster-based wireless sensor networks,” *Measurement: Sensors*, vol. 24, no. May, pp. 100 403,.

[18] A. Zanella, “Internet of things for smart cities,” *IEEE Internet Things J*, vol. 1, no. 1, pp. 22–32,.

[19] A. Al-Mousa and H. Saleh, “An intelligent iot-based architecture towards efficient healthcare facilities,” *International Journal of Computing and Digital Systems*, vol. 12, no. 1, pp. 159–169,.

[20] E. Systems, “Esp32 series.”

[21] J. Ali and M. Zafar, “Improved end-to-end service assurance and mathematical modeling of message queuing telemetry transport protocol based massively deployed fully functional devices in smart cities,” *Alexandria Engineering Journal*, vol. 72, pp. 657–672,.

[22] D. Thakur, Y. Kumar, and S. Vijendra, “Smart irrigation and intrusions detection in agricultural fields using i.o.t,” *Procedia Comput Sci*, vol. 167, pp. 154–162,.

[23] M. Márquez-Vera, M. Martínez-Quezada, R. Calderón-Suárez, A. Rodríguez, and R. Ortega-Mendoza, “Microcontrollers programming for control and automation in undergraduate biotechnology engineering education,” *Digital Chemical Engineering*, vol. 9, no. September, pp. 100 122,.

[24] E. Nwankwo, M. David, and E. Onwuka, “Integration of mqtt-sn and coap protocol for enhanced data communications and resource management in wsns,” *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 3, pp. 1613–1620,.

[25] B. Mishra and A. Kertész, “The use of mqtt in m2m and iot systems: A survey,” *Ieee Access*, vol. 8, pp. 201 071–201 086,.

[26] DFRobot, “Capacitive soil moisture sensor,” pp. 1–6,.

[27] H. M. Co, “Datasheet: Rfm95/96/97/98(w) v1.0,” pp. 121,.

[28] R. Saha, S. Biswas, S. Sarmah, S. Karmakar, and P. Das, “A working prototype using ds18b20 temperature sensor and arduino for health monitoring,” *SN Comput Sci*, vol. 2, no. 1, pp. 33,.

[29] R. semiconductor, “Datasheet bh1750fvi,” pp. 21,.

[30] M. Wurfel, *Temperature and humidity module: DHT11 product manual*. Aosong Guangzhou Electronics Co., Ltd.

[31] M. Tom, “Mqtt-sn protocol - a review.”

[32] J. Alfonso, J. Rodriguez-Fortun, C. Bernad, V. Beliautsou, V. Ivanov, and J. Castellanos, “Geographically distributed real-time co-simulation of electric vehicle.”

[33] M. A. Moon, M. Kaimujjaman, M. M. Hossain, M. M. Islam, and M. S. Hossain, “Seamless real-time thermal imaging system with esp8266: Wireless data transfer and display using udp,” *SN Appl Sci*, vol. 5, no. 11.

[34] J. Lissah and M. Mbise, “Rate adaptive congestion control us-

ing lookup table scheme to enhance quality of experience," pp. 701–711,.

[35] A. Baretto, N. Pudussery, V. Subramaniam, and A. Siddiqui, "Real-time webrtc based mobile surveillance system," *International Journal of Engineering and Management Research*, vol. 11, no. 3.

[36] R. Hellbach, K. Klein, K. Hribernik, and K. Thoben, "Iot-enabled communication systems in testing environments," *Procedia Manuf*, vol. 52, pp. 85–88,.

[37] M. Amaran, N. Noh, M. Rohmad, and H. Hashim, "A comparison of lightweight communication protocols in robotic applications," *Procedia Comput Sci*, vol. 76, no. Iris, pp. 400–405,.

[38] M. Saban, O. Aghzout, L. Medus, and A. Rosado, "Experimental analysis of iot networks based on lora/lorawan under indoor and outdoor environments: Performance and limitations," *IFAC PapersOnLine*, vol. 54, no. 4, pp. 159–164,.

[39] M. McIntosh, "Deployment of a lora-wan near-real-time precision ranching system on extensive desert rangelands: What we have learned*," *Applied Animal Science*, vol. 39, no. 5, pp. 349–361,.

[40] H. Nurwarsito and A. Kusuma, "Development of multipoint lora communication network on microclimate datalogging system with simple lora protocol," in *Proceeding - ICERA 2021: 2021 3rd International Conference on Electronics Representation and Algorithm*, pp. 155–160,.

[41] H. Nurwarsito, K. Adam, C. Prayogo, and S. Oakley, "Development of multi-point lora network with lorawan protocol," 2021, pp. 225–230.

[42] A. Cabuk, "Experimental iot study on fault detection and preventive apparatus using node-red ship's main engine cooling water pump motor," *Eng Fail Anal*, vol. 138, no. April, pp. 106 310,.

[43] W. Hamdy, A. Al-Awamry, and N. Mostafa, "Warehousing 4.0: A proposed system of using node-red for applying internet of things in warehousing," *Sustainable Futures*, vol. 4, no. April, pp. 100 069,.

[44] J. Fiaidhi and S. Mohammed, "Virtual care for cyber–physical systems (vh_cps): Node-red, community of practice and thick data analytics ecosystem," *Comput Commun*, vol. 170, no. February, pp. 84–94,.

[45] W. Jabbar, "Development of lorawan-based iot system for water quality monitoring in rural areas," *Expert Syst Appl*, vol. 242, pp. 122 862,.

[46] M. Hasan, "Building an iot temperature and humidity forecasting model based on long short-term memory (lstm) with improved whale optimization algorithm," *Memories - Materials, Devices, Circuits and Systems*, vol. 6, no. August, pp. 100 086,.

[47] S. Mane, N. Das, G. Singh, M. Cosh, and Y. Dong, "Advancements in dielectric soil moisture sensor calibration : A comprehensive review of methods and techniques," *Comput Electron Agric*, vol. 218, no. December 2023, pp. 108 686,.

[48] B. Li, "Accuracy calibration and evaluation of capacitance-based soil moisture sensors for a variety of soil properties," *Agric Water Manag*, vol. 273, no. August, pp. 107 913,.

[49] Y. Satoh and H. Kakiuchi, "Calibration method to address influences of temperature and electrical conductivity for a low-cost soil water content sensor in the agricultural field," *Agric Water Manag*, vol. 255, no. September 2020, pp. 107 015,.

[50] M. Islam, K. Oliullah, M. Kabir, M. Alom, and M. Mridha, "Machine learning enabled iot system for soil nutrients monitoring and crop recommendation," *J Agric Food Res*, vol. 14, no. June, pp. 100 880,.

[51] L. Spinelle, M. Gerboles, M. Villani, M. Aleixandre, and F. Bonavitacola, "Field calibration of a cluster of low-cost available sensors for air quality monitoring. part a: Ozone and nitrogen dioxide," *Sens Actuators B Chem*, vol. 215, pp. 249–257,.

[52] J. Duarte, D. Noe, and C. Nuñez, "Low-cost soil moisture sensor calibration," pp. 132–142,.

[53] J. Speiran and E. Shakshuki, "Understanding the effect of physical parameters on packet loss in veins vanet simulator," *Procedia Comput Sci*, vol. 201, no. C, pp. 359–367,.