# Delta Bin: Efficient Binary Data Serialization for Low-Power IoT Systems

Payakattu Lahari[1], Avishkar Raut [2],Manitha P V [3]
Department of Electrical and Electronics Engineering
Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, India
Email: bl.en.p2ebs25013@bl.students.amrita.edu[1], bl.en.p2ebs25004@bl.students.amrita.edu[2], pv_manitha@blr.amrita.edu[3]

***Abstract*—In recent years, due to the rapid development of the Internet of Things technologies, the implementation of real-time observation and intelligent decision making can be seen in the field of environmental applications, industry, and real-time sensing. However, current IoT technologies periodically transfer sensor readings, regardless of any changes within the environment, leading to a great amount of communication overhead, bandwidth wastage, and high energy consumption. This paper proposes a DeltaBin inspired selective sensor transmission scheme for efficient energy-aware IoT monitoring systems. The developed approach will employ ESP32-based sensor nodes that are equipped with DHT11 and ultrasonic sensors for temperature and distance monitoring. To facilitate different communication scenarios, this solution uses MQTT and HTTP protocols for transmitting the data from the sensors as well as for interacting with the gateway. In contrast to other schemes that use periodic communication, the gateway-assisted decision making technique that compares the currently obtained values with those that were previously transferred will be used. Under stable operation conditions, constant values are going to stay on the gateway without further generation of messages. Whenever changes occur, the new information will be delivered to the gateway and further synchronized with the cloud visualization service.The suggested solution will increase communication effectiveness and decrease unnecessary transmission costs while allowing scalability of deployment in future IoT applications using low power.**

***Index Terms*—Internet of Things (IoT), ESP32, DeltaBin, Selective Transmission, Gateway Reconstruction, MQTT, HTTP, Real-Time Monitoring, Sensor Data Processing, Cloud Visualization, Low-Power IoT, Efficient Communication**

## I. INTRODUCTION

The Internet of Things (IoT) is one of the innovative technologies that facilitate the communication between devices via sensing and processing mechanisms. Owing to its ability for embedding systems, wireless communication, and cloud technologies, IoT technology has found extensive application in various fields such as environmental monitoring, industrial automation, healthcare applications, smart agriculture, transportation, and intelligent infrastructures.

Current IoT technologies rely significantly on sensing and communication for monitoring and providing operational knowledge. Sensors deployed within the IoT environment capture environmental data continuously and communicate with the central processing platforms or cloud services. While sensing provides intelligence to the system, the periodic transmission of complete packets of sensor data poses many challenges like communication overheads, inefficient bandwidth use, increased power consumption, and lower scalability. Most of the time, IoT devices function in constrained environments where there is little battery storage capacity, memory availability, limited computing power, and lower bandwidth communication links. Transmitting complete data even when no changes occur within the sensor readings leads to wastage of communication resources.

In order to overcome these issues, research has been done in efficient data transmission schemes in IoT communication networks. Some of the data serialization schemes include JSON, BSON, CBOR, Protocol Buffers, and binary encoding to minimize payload size during data transmissions. Despite all these methods, the majority of them still transfer whole sensor readings at specific intervals without considering the changes in the environment. As a result, the idea of delta bin is proposed as one of the most effective binary data transmission techniques because delta bin transmits only the difference between two consecutive sensor observations. In other words, there is no need to transmit full information since the difference will suffice. The main goal of this work is to implement a real-world scenario of a DeltaBin IoT communication protocol using distributed ESP32 sensor nodes along with DHT11 and ultrasonic sensors. The main communication protocols used for data transmission in this work include MQTT and HTTP communications.

Unlike conventional periodic transmission methods, the proposed implementation introduces gateway-assisted selective transmission and reconstruction. Sensor readings are continuously monitored and compared with previously transmitted observations. When sensor values remain unchanged, the gateway preserves and displays the latest available information without initiating additional communication. Whenever variation is detected, only updated values are transmitted and reconstructed before synchronization with cloud visualization. The developed architecture integrates cloud-based monitoring through ThingSpeak to provide real-time observation of sensor behavior. The proposed approach demonstrates efficient communication management while maintaining continuous monitoring performance.

The major contributions of this work are summarized as follows:

- Implementation of selective sensor transmission using DeltaBin-inspired communication principles.
- Development of distributed ESP32-based sensor architecture for environmental monitoring.
- Development of distributed ESP32-based sensor architecture for environmental monitoring.
- Integration of heterogeneous communication protocols using MQTT and HTTP.
- Introduction of gateway-assisted reconstruction for maintaining continuous display of constant values.
- Real-time visualization of reconstructed sensor information using cloud-based monitoring.

## II. LITERATURE SURVEY

Neda Maleki et al., in ”DeltaBin: An Efficient Binary Data Format for Low Power IoT Devices,” have proposed an efficient binary serialization technique for decreasing communication overhead in constrained IoT devices. Several formats have been analyzed in the study such as JSON, BSON, and CBOR to introduce DeltaBin as a selective communication method that transmits only new data instead of continuously sending full sensor packets. Experimental analysis indicated reduced packet size, better transmission speed, and increased efficiency of low-power IoT devices.

A. Al-Fuqaha et al., in ”Internet of Things: A Survey on Enabling Technologies, Protocols and Applications,” have discussed a detailed survey of IoT technologies, communication protocols, and applications of the technology in practical systems. Sensing architecture, network communication, interactions with cloud computing, and various communication protocols for IoT have been discussed in the paper. Efficient protocol selection is important for scalable and reliable IoT communication.

D. Friesel and O. Spinczyk, in ”Data Serialization Formats for the Internet of Things,” have discussed different techniques of data serialization used for IoT communications. Different approaches using both textual and binary formats have been analyzed in the study for reducing communication overheads in IoT. Efficient data transmission has been achieved through experimental results. C. Bormann and P. Hoffman introduced through RFC 8949: Concise Binary Object Representation (CBOR), a compact binary format used to represent data in constrained devices and low bandwidth networks. CBOR showed higher efficiency than traditional text-based models for communications and was instrumental in efficient communication in IoT networks.

Google introduced through Protocol Buffers: Google Data Interchange Format an efficient data interchange format for efficient communication by reducing communication overhead in distributed systems. Protocol Buffers were effective because of their compact format and fast processing speeds.

D. Merriman et al. introduced through BSON Specification Version 1.1, a binary format representing JSON for exchanging structured information. The use of BSON helped improve communication efficiency while being compatible with the traditional formats used in cloud communication systems.

A. Banks and R. Gupta through MQTT Version 3.1.1 Specification proposed a light-weight messaging publish/subscribe framework for IoT sensors. This framework showed efficient message delivery and less network overheads thus allowing deployment of many sensors in real-time.

MathWorks through ThingSpeak IoT Analytics Platform Documentation suggested a web-based analytics and visualization framework for IoT systems. This framework allows sensor data to be collected, dashboards generated, and visualizations created.

M. Aazam and E. Huh in “Fog Computing and Smart Gateway Based Communication for IoT Applications” presented the concept of fog computing which included distributed computing at smart gateways located between IoT sensors and cloud services.

H. Gupta et al. in their article ”Edge Computing for IoT Systems: Architecture and Challenges” talked about edge computing in relation to processing sensor information closer to the source of that information. Advantages were noted in terms of minimized latency, effective use of bandwidth, and scalability.

M. Collina et al. in their article ”Internet of Things Application Layer Protocols Analysis” examined communication protocols applied in IoT implementation, such as MQTT, HTTP, and CoAP. Differences in terms of effectiveness of communication processes, overheads, and applicability of protocols to low-powered embedded systems were demonstrated.

R. Sokullu et al. in their article ”IoT Based Precision Monitoring System Using Sensor Networks” described their IoT architecture for monitoring based on sensing devices and cloud computing. Experiments confirmed effectiveness of the suggested approach.

## III. PROPOSED SYSTEM

The design involves an intelligent IoT architecture with a combination of selective transmissions through sensors with gateways and retransmissions through cloud visualization. The main objective of this proposed architecture is continuous monitoring of environmental factors in order to minimize any form of unwanted transmission between sensing devices and cloud visualization tools.

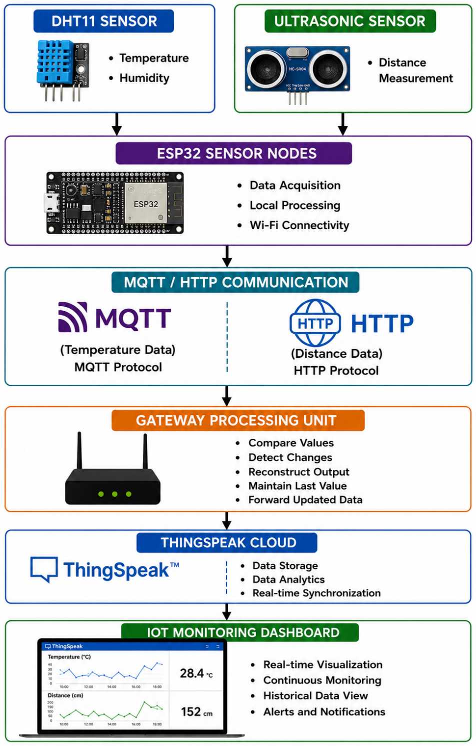

Fig. 1. Proposed IoT Monitoring System Architecture based on ESP32 Sensor Nodes, MQTT/HTTP Communication, Gateway Processing, and ThingSpeak Cloud Visualization.

This architecture involves the use of distributed ESP32 sensor nodes, communication, gateway operations, and cloud visualization. The ESP32 nodes continuously capture physical observations from the environment while carrying out pre-processing prior to transmission.

Differing from typical IoT monitoring architectures that entail periodic transmission of full sensor observations whether there is variation or not, the present architecture involves making decisions on gateway operations involving comparison of sensor readings with previous sensor observations. When there is no change in the environment, then no new sensor observation will be made since the same value will continue to be displayed at the gateway without transmission of new communication packets. The entire process starts with the continuous acquisition of data collected from environmental sensors deployed around. Every ESP32 constantly measures data collected by sensors and sends them to be evaluated at the gateway level.

Data that have been recorded in the past are stored in the gateway, which then does a comparison with the new measurement. When there is no difference between two consecutive measurements, no communication takes place; hence, the previous data will persist in the monitoring view. If there is a detectable change in the measurement, it will be communicated immediately via the communication protocol.

## IV. METHODOLOGY

The proposed approach uses distributed sensing, selective communication, processing with the help of gateways, and visualization with the help of clouds. The designed framework can be used for effective environmental monitoring by ensuring minimal communication due to event-based data transmission.

The proposed system starts with the data gathering process performed by the distributed sensor nodes and ends with the visualization performed via cloud dashboards. Sensor data will be processed by gateway logic before being visualized.

The entire process is shown in Fig. 2. The proposed architecture consists of six operational layers that collectively perform sensing, communication, processing, reconstruction, and visualization.

*1) Sensor Layer:* The sensing layer consists of DHT11 and ultrasonic sensors connected to ESP32 sensor nodes. The DHT11 sensor continuously measures environmental temperature values, while the ultrasonic sensor acquires distance measurements using echo-based sensing principles. Collected measurements are forwarded to the embedded controller for processing.

*2) ESP32 Sensor Node Layer:* SP32 acts as the local processing and communication controller.

Major operations performed include:

- Sensor interfacing
- Data acquisition
- Signal processing
- Wireless communication
- Gateway synchronization

ESP32 enables lightweight IoT deployment through integrated Wi-Fi communication.

*3) Communication Layer:* Sensor information is transmitted using two communication mechanisms. MQTT protocol is implemented for temperature communication because of its lightweight publish–subscribe behavior. HTTP protocol is used for distance data transfer through request–response interaction.

*4) Gateway Processing Layer:* The gateway performs intelligent decision processing.

- Receive sensor measurements
- Extract measurement values
- Compare previous observations
- Detect environmental changes
- Maintain previous values during stable conditions
- Generate update during variations

If values remain unchanged, the gateway continues displaying existing measurements without initiating repeated communication. If changes are detected, updated values are immediately forwarded. The communication layer enables reliable transmission between sensor nodes and gateway processing.

*5) Cloud Layer:* ThingSpeak is used as the cloud monitoring platform. Cloud operations include:

- Data collection
- Real-time synchronization
- Historical storage
- Monitoring support

Only updated measurements are synchronized to reduce communication redundancy.

*6) Dashboard Layer:* The dashboard provides continuous observation of reconstructed gateway values.

- Temperature monitoring
- Distance monitoring
- Historical trends
- Real-time updates

The dashboard updates immediately whenever gateway reconstruction generates a new observation.

### *A. Hardware Implementation*

Hardware Implementation:For the development of IoT monitoring solution hardware components such as sensor-based devices connected through the ESP32 were used for real-time collection and processing of environmental data. The full hardware architecture includes ESP32 controllers, DHT11 temperature sensors, ultrasonic sensors, gateway processors, and visualization interfaces in the cloud.

ESP32 acts as an embedded processor for performing sensor operations and communication with other modules. Due to its built-in Wi-Fi functionality, compact design, and efficient consumption of electrical energy, ESP32 is an effective platform to develop efficient IoT-based systems. This component processes sensor information, preprocesses the obtained readings, and initiates the communication process. Environmental temperature measurement is done by utilizing the DHT11 sensor. The component regularly detects surrounding temperature values and produces a digital signal, which then gets processed in the ESP32. The DHT11 sensor provides efficient measurement of environmental parameters while maintaining relatively low computational and power needs. The process of distance monitoring involves the application of an ultrasonic sensor working based on the principle of ultrasonic waves that transmit and reflect from objects. The sensor measures the time taken by ultrasonic waves in the process of propagation to determine the distance. Distance is calculated depending on the period taken by the waves and the results are transferred to the communication module.

A gateway processing module is embedded in order to enhance the process of communication and intelligence of the data exchange. The gateway receives the sensor information, keeps previous records, compares the information, constructs the output results and updates the monitoring window. In situations where there is stable environment, the recorded data will continue being shown without any need for further communication until a change occurs.

In case changes occur in the sensor information, the new data will be received and processed in real time by the visualization software.

### *B. Software Implementation*

The design of the software implementation of the proposed system was performed with respect to the embedded sensing, communication management, gateway process management, and cloud-based visualization purposes. The software architecture consists of the execution of firmware, communication, selective data transmission, and monitoring capabilities.

Embedded firmware coding was performed with the help of the Arduino IDE environment, which ensures a high-quality coding experience for the ESP32 microcontroller. Firmware code includes initialization, measuring data acquisition and processing, communication management, and synchronization processes with gateway and cloud nodes. Continuous monitoring and evaluation of sensor measurements are performed within the runtime. The proposed solution incorporates both MQTT and HTTP communications for different types of tasks. MQTT communication is chosen for transferring the temperature measurements based on the lightweight publish-subscribe model and lower communication overhead. HTTP communication is applied for distance transmission based on request-response approach.

The key software innovation for the proposed work is the use of gateway-based selective transmission functionality. The incoming sensor readings are constantly compared to the previous sensor readings that were transmitted earlier. If there is no change observed between two successive measurements, then no transmission takes place, and the old value persists in the gateway until the variation in the environment triggers transmission.

The cloud connectivity is made possible by using ThingSpeak visualization services, providing the capability for live dashboard monitoring, observing past values, and displaying the reconstructed readings. Hence, the proposed software architecture provides effective communication capability along with monitoring functionality.

## V. EXPERIMENTAL ANALYSIS

To evaluate the efficiency of the suggested selective sensor transmission scheme under practical operating scenarios, experimental validation was performed. To this end, the developed scheme was realized using a combination of ESP32 sensor nodes embedded with DHT11 and ultrasonic sensors for environmental monitoring and measuring distance purposes, respectively. In total, the realized scheme comprised the following stages: sensors acquisition, communication, gateway processing, and visualizing in the cloud environment of ThingSpeak. Communication between the sensors and gateway was carried out via MQTT and HTTP schemes.Communication performance and gateway reconstruction ability have been evaluated as the main aspects of the experiment. The implemented test environment included:

TABLE I
EXPERIMENTAL SETUP CONFIGURATION

| **Component** | **Configuration** |
|---|---|
| Controller | ESP32 |
| Temperature Sensor | DHT11 |
| Distance Sensor | Ultrasonic |
| Communication | MQTT + HTTP |
| Processing | Gateway Reconstruction |
| Visualization | ThingSpeak |
| Latency | 80–150 ms (MQTT), 120–250 ms (HTTP) |

### A. Experimental Procedure

The experimental procedure was divided into two operating conditions to evaluate system behavior.

*1) Case 1: Stable Sensor Condition:* Sensor values remained relatively constant for an extended period.

The gateway continuously monitored incoming observations and maintained previously displayed values without generating repeated communication.

The objective of this test was to validate selective transmission and communication reduction.

*2) Case 2: Dynamic Sensor Condition:* Sensor values were intentionally varied to simulate environmental changes.

Whenever updated measurements were detected:

- Communication was initiated
- Gateway reconstruction was performed
- Dashboard values were refreshed

This experiment validated event-driven transmission.

### B. Experimental Observations

The obtained observations demonstrated successful implementation of gateway-assisted selective transmission.

Temperature measurements obtained from DHT11 remained relatively stable throughout the monitoring period and showed only minor fluctuations around the measured range.

Distance measurements acquired from the ultrasonic sensor exhibited dynamic variation and generated communication updates only during environmental changes.

During unchanged operating conditions:

- Gateway preserved previous values
- Communication remained inactive
- Dashboard maintained continuous visualization

During changing conditions:

- Updated values were transmitted
- Gateway reconstructed measurements
- Dashboard synchronized automatically

These observations confirm successful implementation of intelligent communication management.

### C. Experimental Results

Experimental dashboard analysis demonstrated successful operation under both stable and dynamic monitoring conditions.

TABLE II
OBSERVED EXPERIMENTAL RESULTS

| **Parameter** | **Observed Range** |
|---|---|
| Temperature | 28°C – 28.2°C |
| Distance | 30 cm – 260 cm |
| Communication | MQTT + HTTP |
| Dashboard Response | Real-Time |
| Gateway Processing | Successful |

### D. Comparative Analysis

The proposed architecture was compared with conventional periodic transmission systems.

### E. Performance Discussion

The experimental evaluation demonstrates that the proposed selective transmission framework effectively reduces redundant communication while maintaining continuous monitoring. The integration of gateway reconstruction enabled stable

TABLE III
COMPARISON BETWEEN CONVENTIONAL AND PROPOSED SYSTEM

| **Parameter** | **Conventional System** | **Proposed System** |
|---|---|---|
| Communication | Continuous | Selective |
| Updates | Periodic | Event Driven |
| Gateway Processing | Basic | Intelligent |
| Dashboard Refresh | Repeated | Conditional |
| Network Load | Higher | Reduced |
| Monitoring Efficiency | Moderate | Improved |

visualization without repeated packet transmission. Real-time updates occurred only during environmental changes, improving communication efficiency and reducing unnecessary processing overhead.The results validate the applicability of the proposed architecture for low-power and scalable IoT monitoring systems.

## VI. RESULTS AND DISCUSSION

The proposed IoT monitoring system was experimentally validated using real-time sensing, gateway processing, and cloud visualization. Experimental evaluation was performed to verify selective transmission, gateway reconstruction, and dashboard synchronization under different environmental conditions.

The implemented architecture successfully acquired temperature and distance measurements using ESP32 sensor nodes and transmitted updates through MQTT and HTTP communication.

### A. *Dashboard Visualization Results*

Fig. **??** shows the real-time visualization obtained from the ThingSpeak monitoring platform. The dashboard continuously displayed reconstructed sensor values generated at the gateway layer. Temperature monitoring remained stable throughout the observation period and showed minor variations around 28°C, indicating stable environmental behavior.

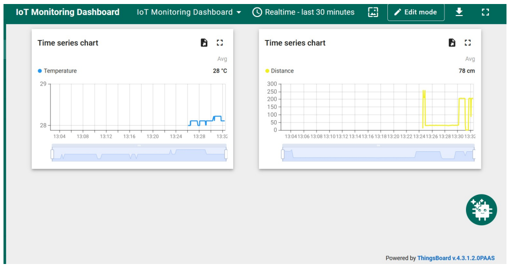

Fig. 2. Real-Time Dashboard Visualization using ThingSpeak

Distance measurements exhibited dynamic response characteristics and generated immediate updates whenever environmental changes occurred. The dashboard updated automatically after receiving modified sensor information.

### B. *Gateway Monitoring Results*

Fig. 3 illustrates the gateway-side monitoring and visualization output.

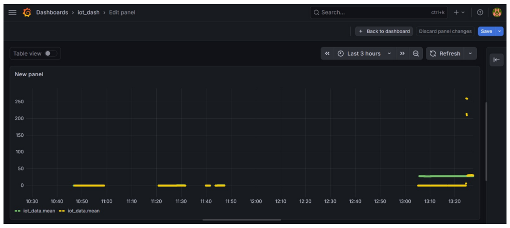

Fig. 3. Gateway Monitoring and Selective Data Update Visualization

The gateway maintained continuous display of previous values during stable operating conditions and generated transmission only when variations were detected. The obtained output confirms successful implementation of gateway-assisted selective transmission.

### C. *Observed Experimental Results*

TABLE IV
OBSERVED SYSTEM PERFORMANCE

| **Parameter** | **Observed Value** |
|---|---|
| Temperature | 28°C – 28.2°C |
| Distance | 30 cm – 260 cm |
| Communication | MQTT + HTTP |
| Gateway Processing | Selective Transmission |
| Dashboard Response | Real-Time |
| Cloud Visualization | Successful |

### D. *Comparative Performance Analysis*

TABLE V
COMPARISON BETWEEN CONVENTIONAL AND PROPOSED SYSTEM

| **Parameter** | **Conventional** | **Proposed** |
|---|---|---|
| Communication | Continuous | Selective |
| Updates | Periodic | Event Driven |
| Gateway | Basic | Intelligent |
| Dashboard Refresh | Repeated | Conditional |
| Network Load | Higher | Reduced |
| Efficiency | Moderate | Improved |

### E. *Discussion*

Experimental observations demonstrate that the proposed architecture successfully reduces unnecessary communication while maintaining continuous monitoring capability.

Under stable environmental conditions, gateway reconstruction preserved previously received values and prevented repeated transmission. During dynamic conditions, updated measurements were immediately communicated and visualized.

The obtained results validate that selective transmission combined with gateway-assisted reconstruction improves communication efficiency, reduces network overhead, and supports scalable IoT monitoring applications.

## VII. CONCLUSION

This paper presented an intelligent IoT monitoring framework based on selective sensor data transmission and gateway-assisted reconstruction for efficient real-time monitoring. The proposed system was developed using ESP32 sensor nodes integrated with DHT11 and ultrasonic sensors to acquire environmental measurements and transmit information through MQTT and HTTP communication mechanisms. Unlike conventional IoT architectures that continuously transmit sensor readings irrespective of environmental conditions, the proposed approach introduced gateway decision processing to minimize unnecessary communication. The gateway continuously compared current measurements with previously received observations and maintained display of constant values without repeated transmission. Whenever environmental changes were detected, updated measurements were immediately transmitted and synchronized with cloud visualization.

Experimental evaluation demonstrated successful acquisition, communication, reconstruction, and real-time dashboard monitoring using ThingSpeak. The obtained results confirmed stable operation under both constant and dynamic sensing conditions while reducing communication redundancy and improving monitoring efficiency. The integration of selective transmission, intelligent gateway processing, and cloud-based visualization establishes an efficient and scalable architecture for next-generation IoT monitoring applications. The developed framework can be extended for large-scale environmental monitoring and smart sensing systems requiring optimized communication and continuous observation.

**Future Scope** : Future improvements may include integration of multiple distributed sensor nodes, implementation of edge intelligence for predictive decision making, incorporation of machine learning algorithms for anomaly detection, and deployment of advanced cloud analytics for large-scale monitoring environments. Additional optimization of communication protocols and energy-efficient gateway processing may further enhance overall system performance.

## REFERENCES

[1] Islam, SM Riazul, Daehan Kwak, MD Humaun Kabir, Mahmud Hossain, and Kyung-Sup Kwak. ”The internet of things for health care: a comprehensive survey.” IEEE access 3 (2015): 678-708.

[2] Gündogan, C., P. Kietzmann, T. C. Schmidt, and M. Wählisch. ”ICNLoWPAN—Named-Data Networking for Low Power IoT Networks. arXiv 2018.” arXiv preprint arXiv:1812.07025.

[3] He, Wenjie, Bo Chu, Rui Gu, Haitao Zhang, Bin Shan, and Rong Chen. ”The application of generalized predictive control in the radiant heating atomic layer deposition reactor.” In 2013 IEEE International Symposium on Assembly and Manufacturing (ISAM), pp. 37-40. IEEE, 2013.

[4] Ziółkowski, Cezary, and Jan M. Kelner. ”Statistical evaluation of the azimuth and elevation angles seen at the output of the receiving antenna.” IEEE Transactions on Antennas and Propagation 66, no. 4 (2018): 2165-2169.

[5] Silva, Eduardo, Alfredo Martins, André Dias, Aníbal Matos, Augustin Olivier, Carlos Pinho, Filipe Aranda de Sá et al. ”Strengthening marine and maritime research and technology.” In OCEANS 2016 MTS/IEEE Monterey, pp. 1-9. IEEE, 2016.

[6] Shi, Weisong, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. ”Edge computing: Vision and challenges.” IEEE internet of things journal 3, no. 5 (2016): 637-646.

[7] Serepas, Filippos, Ioannis Papias, Konstantinos Christakis, Nikos Dimitropoulos, and Vangelis Marinakis. ”Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller.” Computers 14, no. 9 (2025): 391.

[8] Soni, Dipa, and Ashwin Makwana. ”A survey on mqtt: a protocol of internet of things (iot).” In International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017), vol. 20, no. April, 2017. 2017.

[9] Chittem, VishnuVardhan, Sai Karthik Mukunda, and Eshwar Sai Reddy Mudumala. ”Smart Saffron Cultivation: An IoT-Based Environmental Monitoring and Control System with GSM Alert Integration.” In 2025 10th International Conference on Communication and Electronics Systems (ICCES), pp. 665-669. IEEE, 2025.

[10] Garikipati, Karthik, Tejaswi Muppala, A. Vinitha Chowdary, and Apurvanand Sahay. ”IoT sensor data stream compression with hybrid compression algorithms.” In 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1-8. IEEE, 2024.

[11] Ramesh, S. R. ”Pipelined and Low-Power MAC Unit with Sparsity-Aware Encoding for Neural Network Applications.” In 2025 IEEE 14th International Conference on Communication Systems and Network Technologies (CSNT), pp. 1210-1215. IEEE, 2025.

[12] Vidhya, S. S., Senthilkumar Mathi, V. Ananthanarayanan, and Ganesh Neelakanta Iyer. ”Ip-rpl: An intelligent power-aware routing protocol for next-generation low-power networks.” IEEE Sensors Journal 25, no. 2 (2024): 3640-3648.

[13] Tejesh, B., G. Udaya Baskar, B. Nanda Krishna, K. Deepa, P. V. Manitha, and V. Sailaja. ”A comparative analysis of power generation using solar and fuel cell for charging EVs.” In 2022 International Virtual Conference on Power Engineering Computing and Control: Developments in Electric Vehicles and Energy Sector for Sustainable Future (PECCON), pp. 1-5. IEEE, 2022.

[14] Praveenraja, T., and P. V. Manitha. ”Integration of IoT-Based Air and Sound Pollution Monitoring with HVAC System for Enhanced Indoor Environment Management.” In 2025 5th International Conference on Expert Clouds and Applications (ICOECA), pp. 332-338. IEEE, 2025.