Design and Development of a Decentralized Fault Detection System for Street Lights Using IoT

Divya Dudure
Department of Industrial IoT
St. Vincent Pallotti College of Engineering and Technology
Nagpur, Maharashtra, India
divyadudure@gmail.com

Apurva Bhongade
Department of Industrial IoT
St. Vincent Pallotti College of Engineering and Technology
Nagpur, Maharashtra, India
apurvab0205@gmail.com

Prajwal Kamble
Department of Industrial IoT
St. Vincent Pallotti College of Engineering and Technology
Nagpur, Maharashtra, India
prajwaltkamble9373@gmail.com

Sanved Latkar
Department of Industrial IoT
St. Vincent Pallotti College of Engineering and Technology
Nagpur, Maharashtra, India
sanved5latkar@gmail.com

Rohan Vaidya
Department of Industrial IoT
St. Vincent Pallotti College of Engineering and Technology
Nagpur, Maharashtra, India
rvaidya@stvincentngp.edu.in

Abstract

Urban infrastructure management is a cornerstone of smart city development, with streetlight systems representing a significant operational cost and public safety responsibility. Traditional streetlight maintenance relies on manual inspections and public complaints, leading to delayed repairs, energy wastage, and compromised pedestrian safety. This paper presents a low-cost, scalable, and decentralized Internet of Things (IoT) framework for the real-time monitoring and fault detection of urban streetlights. The system employs a novel dual-sensor approach at each streetlight (edge node) to autonomously diagnose its operational state by comparing ambient light with its own luminaire output. This allows for the precise detection of two primary faults: lights ON during the day (energy waste) and lights OFF during the night (safety hazard). Each edge node, built around an ESP32 microcontroller, transmits its status via a LoRa (Long Range) communication module to a central gateway. The gateway aggregates data and relays it to a Firebase Realtime Database. A web-based dashboard provides a centralized, live view of the entire network's health, with intuitive visual indicators for normal, fault, and offline conditions. The decentralized architecture ensures high scalability and resilience, while the use of low-power components and long-range communication makes the system economically viable for large-scale municipal deployments. Experimental results from a prototype deployment demonstrate high reliability with under 2% packet loss and a fault-to-dashboard latency of under 10 seconds.

Index Terms

Internet of Things (IoT), Smart Streetlight, Fault Detection, LoRa, Edge Computing, ESP32, Firebase, Smart City.

I. INTRODUCTION

The rapid pace of urbanization presents significant challenges for municipal authorities, who are tasked with managing complex public infrastructure efficiently and sustainably. Among these systems, public street lighting is one of the most critical. It is essential for ensuring road and pedestrian safety, deterring crime, and supporting nighttime economic activity. However, it also represents a substantial portion of a city's energy consumption and operational budget [1]. The hidden costs of inefficient infrastructure are vast; a case study on traffic congestion in New Delhi estimated annual costs reaching millions of US dollars due to lost productivity and wasted fuel. Similarly, poorly managed streetlight systems incur significant hidden costs through energy waste and the societal impact of reduced public safety. The conventional approach to streetlight management is predominantly reactive, relying on scheduled manual patrols or citizen reports to identify faults. This methodology is inherently inefficient, resulting in prolonged periods where lights are non-operational, posing safety risks, or remain active during daylight hours, leading to significant energy wastage.

The advent of the Internet of Things (IoT) has unlocked new possibilities for transforming traditional urban systems into intelligent, interconnected networks [2]. Smart streetlight solutions have emerged as a prominent application within the smart city paradigm, promising benefits such as remote control, energy optimization through dimming, and automated fault reporting

[3]. However, many existing commercial solutions are either prohibitively expensive, rely on costly cellular communication (GPRS/4G), or are built on centralized architectures that can be difficult to scale and may have single points of failure.

This paper addresses these limitations by proposing a Decentralized Real-Time Fault Detection System for urban streetlights. Our approach is founded on three core principles: decentralization, low cost, and real-time responsiveness. We empower each streetlight with the intelligence to diagnose its own state, a concept we term "edge-level autonomy." By using a dual-sensor configuration, each node can make a definitive assessment of its health without requiring complex server-side analytics. This aligns with the principles of modern IoT architectures where local processing at the edge is preferred to reduce latency and bandwidth use. Communication is handled by LoRa (Long Range) technology, a low-power wide-area network (LPWAN) protocol ideal for smart city applications due to its long range and license-free spectrum operation [4]. This combination of localized intelligence and efficient communication results in a system that is not only highly scalable and resilient but also economically viable for widespread deployment in both developing and developed urban environments. The potential impact of such a system is significant, as summarized in Table I.

The main contributions of this work are:

- A novel dual-sensor fault detection algorithm that runs autonomously on each streetlight node.
- A three-tier IoT architecture leveraging edge computing for fault diagnosis, a gateway for data aggregation, and a cloud backend for remote monitoring.
- The integration of LoRa for robust, low-power, and long-range communication, minimizing operational costs.
- The development and testing of a functional prototype that validates the system's performance in terms of reliability and latency.

This paper is organized as follows: Section II reviews related work in the field of smart streetlight monitoring. Section III details the proposed system architecture. Section IV explains the methodology and algorithms behind the fault detection and data communication. Section V describes the hardware and software implementation in detail. Section VI presents the experimental setup and results. Section VII discusses the implications of the results, and Section VIII concludes the paper with an outlook on future work.

TABLE I
COMPARISON OF STREETLIGHT METRICS BEFORE AND AFTER SYSTEM IMPLEMENTATION.

Metric	**Before Implementation**	**After Implementation**	**Net Change**
Day-Burner Rate	10% of fixtures	≈0% of fixtures	−10 pp[1]
Daytime Energy Waste	2.5–4% of municipal lighting load	0% of municipal lighting load	−2.5–4 pp
Light-Pollution Loss	≥30% of outdoor lighting energy	≤5% of outdoor lighting energy	−25 pp
National Electricity Load (India)	1.5% of national consumption	0.6–1.125% of national consumption	−0.375–0.9 pp
O&M Service Trips	100% manual fault visits	50–70% manual fault visits	−30–50%
Total Lighting Budget	Baseline (100%)	80% of baseline	−20%
Fault-to-Dashboard Latency	Hours to days	<10 s	−99.99%

A decentralized system spreads processing, decisionmaking, and control across multiple independent nodes instead of relying on one central unit. Each node has its own computing power, which lets it analyze data and make decisions locally. In streetlight fault detection, decentralization means each streetlight pole can sense its surroundings, process the data it gathers, and assess its operational status. This method places intelligence at the network's edge, with each pole acting as an independent unit that can detect faults and report them.

II. RELATED WORK

The concept of smart street lighting has been an active area of research for over a decade, with various approaches proposed to enhance efficiency and automate maintenance. Early systems focused primarily on remote switching and dimming capabilities. For instance, the work by Lavric *et al.* explored a Power Line Communication (PLC) based system, which offered reliable control but faced challenges with installation complexity and noise on the power lines [5].

With the maturation of wireless technologies, researchers shifted towards more flexible communication methods. Zigbee and other mesh networking protocols were popular choices for creating localized networks of streetlights. The system proposed by Anvari-Moghaddam *et al.* used a Zigbee mesh network where nodes could relay data to a central coordinator [6]. While effective for dense deployments, these systems often suffered from limited range and complex network management, particularly as the network scaled. The introduction of cellular communication, such as GPRS, provided a more direct-to-cloud approach, as demonstrated by M. T. ERDEM *et al.* [7]. These systems offered simplicity in network architecture but incurred significant operational costs due to SIM card subscriptions and data plans for each node, making them less feasible for large-scale municipal projects.

The emergence of Low-Power Wide-Area Network (LPWAN) technologies, such as LoRa and NB-IoT, marked a significant advancement for smart city applications [8]. LoRa, in particular, has been identified as a strong candidate for streetlight monitoring due to its excellent balance of range, low power consumption, and low cost [9]. Several studies have proposed LoRa-based streetlight systems. The work by Leonardi *et al.* presents a LoRaWAN architecture for controlling and monitoring streetlights, demonstrating its feasibility [10]. Similarly, Callebaut *et al.* proposed a large-scale street lighting control system using LoRaWAN, focusing on network performance and scalability [11]. However, many such systems still rely on a centralized server to perform the fault detection logic. The nodes are often simple sensors that report raw data (e.g., current consumption, LDR value), and the intelligence resides entirely in the cloud. This can introduce latency and creates a dependency on a constant internet connection for the core functionality. Further research has specifically focused on automated fault diagnosis [12], with systems using LoRa to report pre-diagnosed faults [13], and on the specific challenges of fault tolerance in wireless sensor networks for urban lighting [14].

Our proposed system distinguishes itself from the existing literature in two key aspects. First is its emphasis on decentralized fault detection. By equipping each node with two light sensors and the logic to interpret their readings in context, we push the intelligence to the absolute edge of the network. The node does not just report data; it reports a conclusion (e.g., status code '3' for "FAULT: Night, Light OFF"). This reduces data payload, minimizes cloud processing requirements, and allows the node to be aware of its status even if its connection to the gateway is temporarily lost. This edge-centric philosophy is critical for creating responsive and robust systems for unpredictable urban conditions. Second is the supervisory role of the gateway, which actively monitors the network for offline nodes—a critical fault condition that many systems overlook. By maintaining a lastSeen timestamp for each pole, our gateway can proactively identify and report a a complete node failure, which could indicate a power outage or hardware malfunction. This combination of edge-level autonomy and gateway-level supervision, along with a failsafe fallback mechanism, creates a more robust and resilient monitoring framework than previously proposed solutions.

III. PROPOSED SYSTEM ARCHITECTURE

The system is designed based on a hierarchical three-tier IoT architecture, which segregates functionalities for enhanced modularity, scalability, and maintainability. The three tiers are the Edge Tier, the Gateway Tier, and the Cloud Tier.

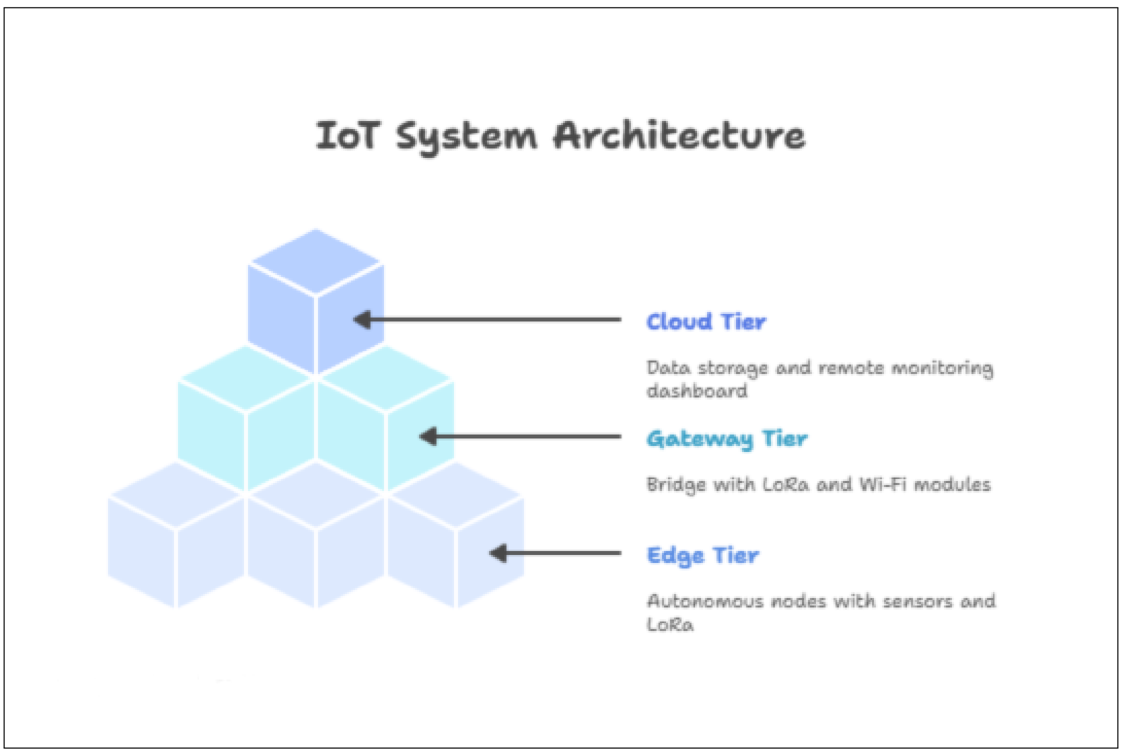

Fig. 1. Block diagram of the 3-tier system architecture, showing the Edge Node, Gateway, and Cloud.

A. Edge Tier (Transmitter Node)

The Edge Tier consists of autonomous transmitter nodes installed on each streetlight pole. This tier is the foundation of the system's decentralized intelligence. Each node is a self-contained unit responsible for sensing its local environment, executing the fault detection algorithm, and transmitting its status. The key components of an Edge Node are:

- **Microcontroller (ESP32):** A powerful, low-cost microcontroller with dual-core processing capabilities and ultra-low-power co-processors [15]. It orchestrates all tasks on the node, from reading sensors to managing the LoRa module and entering deep sleep mode to conserve power.
- **Light Sensors (2x BH1750):** Two high-precision digital light sensors are used.
  - *Ambient Sensor:* Positioned to measure the general ambient light, shielded from the streetlight's own lamp, to accurately determine day or night conditions.
  - *Luminaire Sensor:* Positioned to exclusively measure the light output from the pole's own lamp to determine if it is ON or OFF.
- **Communication Module (SX1278 LoRa):** A long-range transceiver operating in the 433 MHz ISM band [16]. It is used to transmit the node's status to the gateway over distances of several hundred meters to kilometers, depending on the environment.

B. Gateway Tier (Receiver Node)

The Gateway Tier acts as a bridge between the low-power LoRa network of edge nodes and the high-bandwidth internet. A single gateway is designed to receive data from numerous transmitter nodes within its radio range. It functions as a local coordinator, parsing incoming data, supervising node connectivity, and securely relaying structured information to the cloud backend. The key components of the Gateway Node are:

- **Microcontroller (ESP32):** The same powerful microcontroller is used at the gateway for its processing capabilities and integrated Wi-Fi.
- **Communication Module (SX1278 LoRa):** This module is configured in receiver mode to continuously listen for incoming packets from the edge nodes.
- **Wi-Fi Module (Integrated in ESP32):** The built-in Wi-Fi is used to connect the gateway to a local internet access point (e.g., a router) to relay the received data to the cloud.

C. Cloud Tier (Backend & Dashboard)

The Cloud Tier is responsible for data persistence, visualization, and providing a user interface for remote monitoring.

- **Cloud Backend (Firebase Realtime Database):** A cloud-hosted NoSQL database service [17]. It was chosen for its real-time data synchronization capabilities, which allow the dashboard to update instantly as new data arrives from the gateway. Its flexible, schema-less JSON data structure is well-suited for evolving IoT data requirements.
- **Web-Based Dashboard:** A custom front-end application built with HTML, CSS, and JavaScript. It connects securely to the Firebase database and visualizes the status of every streetlight in the network in an intuitive, map-based, or card-based layout. The dashboard provides summary statistics (e.g., total online, faults, offline) and allows operators to see the health of the entire system at a glance.

IV. METHODOLOGY AND ALGORITHMS

The core of the system's intelligence lies in a hybrid protocol that combines autonomous fault detection at each transmitter node with a receiver-managed polling scheme for communication. This ensures that each streetlight can intelligently determine its own status while the network remains efficient and collision-free. This section details the operational logic of the transmitter and gateway nodes.

A. Autonomous Fault Detection at the Edge

Each transmitter node executes a periodic sense-process-transmit cycle.

1) **Sensing:** Upon waking from deep sleep, the ESP32 powers on the two BH1750 sensors and takes luminosity readings: L_{ambient} from the ambient sensor and L_{pole} from the luminaire sensor.
2) **State Determination:** The readings are compared against pre-calibrated thresholds to determine two boolean states:

$$\text{isNight} = (L_{\text{ambient}} < \text{Threshold}_{\text{night}}) \tag{1}$$

$$\text{isLightOn} = (L_{\text{pole}} > \text{Threshold}_{\text{on}}) \tag{2}$$

These thresholds are determined empirically during a calibration phase to suit the local environment. This is critical, as a fixed threshold may not account for seasonal variations in daylight or overcast weather conditions.
3) **Status Code Generation:** The two boolean states are mapped to a single integer `statusCode` for efficient transmission. This code represents the complete health of the streetlight, as shown in Table II.
4) **Transmission and Sleep:** The node formats a minimal LoRa packet as a comma-separated string (e.g., "101,2" for Pole ID 101 with a fault). After successful transmission, the ESP32 deactivates all peripherals and enters a deep sleep mode for a predefined interval (e.g., 5 minutes) to conserve energy.

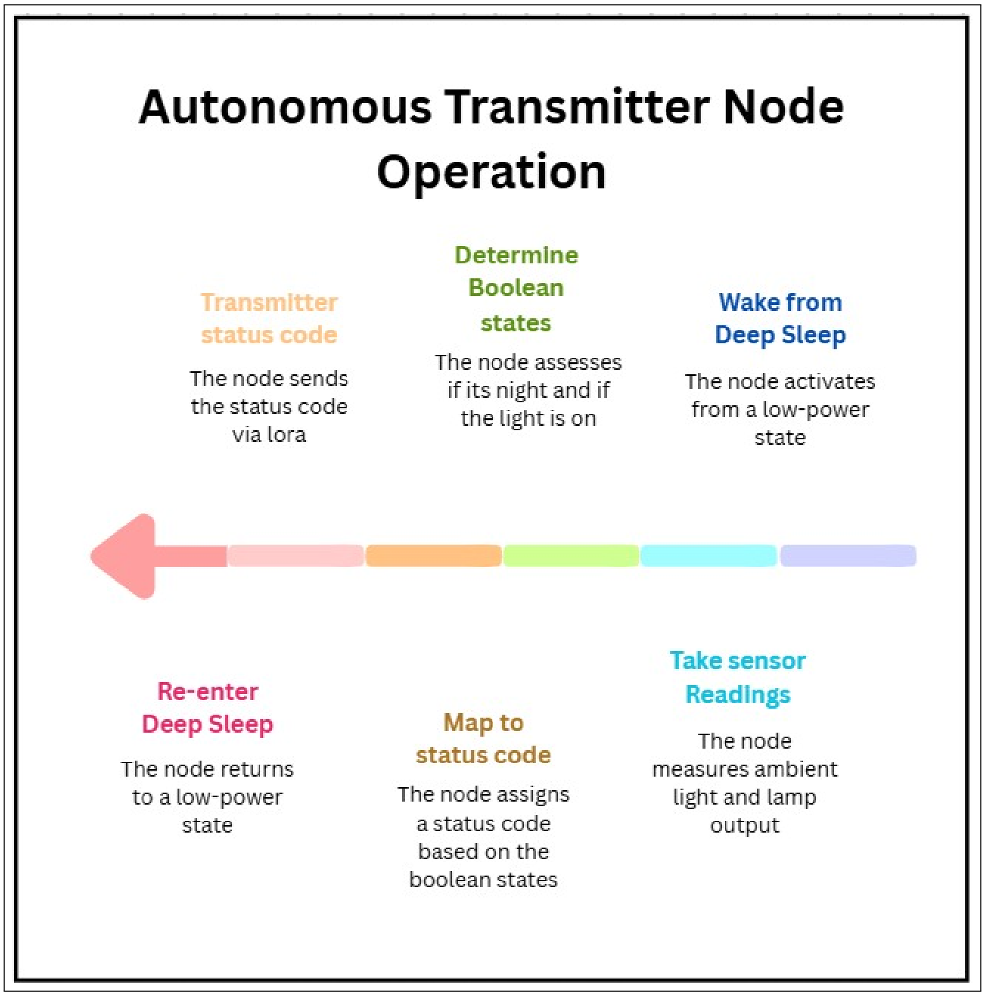

Fig. 2. The Transmitter Node's operational logic, from wake-up to deep sleep.

TABLE II
STATUS CODE MAPPING

isNight	isLightOn	statusCode	Status Description
false	false	0	OK: Day, Light OFF
true	true	1	OK: Night, Light ON
false	true	2	FAULT: Energy Waste
true	false	3	FAULT: Safety Hazard

B. Gateway Coordination and Supervision

The gateway's firmware is designed for continuous, reliable operation, acting as the central coordinator and supervisor for the network.

1) **Sequential Polling and Parsing:** To ensure efficient and collision-free communication, the gateway employs a sequential polling scheme. It manages a short "transmission window" (e.g., 10 seconds) for each pole in sequence. The gateway only listens for a packet from the expected `PoleID` during its window. If a packet is received or the window expires, it shifts its attention to the next pole in the sequence. When a valid packet is received, it is parsed to extract the `poleID` and `statusCode`.
2) **State Management:** The gateway maintains a data structure (e.g., a hash map or array of objects) in its memory for every known pole in its network. This structure stores the latest `statusCode`, an `isOnline` flag, and a `lastSeen` timestamp. Upon receiving a packet, it updates the corresponding pole's entry with the new data and the current time.
3) **Supervisory Monitoring (Offline Detection):** A non-blocking supervisory timer runs periodically on the gateway (e.g., every minute). This timer iterates through all known poles and checks if $(\text{currentTime} - \text{lastSeen}) > \text{Timeout}_{\text{Offline}}$ (e.g.,

15 minutes). If a pole has not been heard from for longer than the timeout period, the gateway proactively marks its status as offline by setting `isOnline = false` and assigning a special status code (e.g., -1). This is a critical feature for detecting complete power or hardware failures at the node.

4) **Cloud Synchronization:** Whenever a new status is received from a node OR a node is marked as offline by the supervisory timer, the gateway formats a structured JSON object and sends it to the Firebase Realtime Database. The data is stored under a path corresponding to the pole's ID (e.g., `/StreetLights/Pole_101`). A typical JSON payload would look like this:

V. Hardware and Software Implementation

The selection of hardware and the design of the software were driven by the goals of low cost, low power consumption, and high reliability. This section provides a detailed justification for the key components chosen.

A. Core Hardware Selection

ESP32 Microcontroller:

- *Edge Node (ESP32 as Sensor Node):* Wakes up from deep sleep. Reads sensor data (e.g., LDR → day/night, light ON/OFF). Processes it locally (detects fault: light ON in day). Sends this data via LoRa/GSM/Wi-Fi depending on setup. Goes back to sleep.
- *Gateway Node (ESP32 as Bridge to Cloud):* Receives packets from edge nodes (via LoRa/UART/etc.). Uses Wi-Fi to push data to the cloud (ThingSpeak, MQTT). Optionally, uses Bluetooth for local maintenance/debugging. Stays more active than edge nodes, but still manages power efficiently compared to a Raspberry Pi.

BH1750 Light Sensor:

- *Edge Node Role (BH1750 + ESP32):* BH1750 constantly "watches" the environment's brightness. Provides direct lux reading to ESP32 via I²C. ESP32 uses this data to: Detect day/night transitions. Check if a streetlight is ON when it shouldn't be (e.g., glowing in daytime). Report abnormal conditions (like lamp failure).
- *Why it's perfect here:* With its precision & wide range, you can detect subtle changes (dusk/dawn) and also handle harsh streetlamp glare. Keeps firmware simple → no manual calibration headaches.

LoRa Technology and the SX1278 Module:

- *Edge Node (with SX1278):* ESP32 reads sensor data (from BH1750 or LDR). Sends data via SPI to SX1278. SX1278 transmits it using LoRa at 433 MHz.
- *Gateway Node (with SX1278):* SX1278 receives LoRa packet. Passes it via SPI to ESP32. ESP32 uploads it to the cloud (ThingSpeak, dashboard).

B. Software, Network, and Backend Implementation

Firmware Development:

- *Edge Node Firmware (ESP32 + BH1750 + SX1278):* Startup: ESP32 wakes from deep sleep. Sensor Acquisition: BH1750 provides calibrated lux values via I²C. Fault Detection: Light ON during daytime → Fault. Light OFF during nighttime → Fault. Data Transmission: Sends pole ID, status, and fault flag via LoRa (SX1278). Power Management: Returns to deep sleep after transmission.
- *Gateway Node Firmware (ESP32 + SX1278 + Wi-Fi + Firebase):* LoRa Listener Task: Continuously receives packets from edge nodes. Offline Detection Timer: Flags a node as "offline" if no packet is received within a set interval. Cloud Communication Task: Updates Firebase in real-time using non-blocking operations.

Firebase Realtime Database:

- *Data Ingestion:* Gateway nodes receive LoRa packets, parse pole ID, lux value, fault flag, and status, and write the data as JSON objects to Firebase.
- *Real-Time Synchronization:* The database instantly pushes updates to all connected clients, enabling live dashboards and mobile apps to display streetlight status, faults, and offline alerts.
- *Scalability:* The schema-less design allows seamless integration of new nodes or additional parameters (e.g., pole temperature) without database migration.
- *System Benefits:* Ensures real-time responsiveness, operational transparency, and long-term adaptability of the monitoring system.

Network Reliability and Configuration: To ensure robust operation in real-world deployment, multiple reliability measures were implemented:

- *Stable Connectivity:* The gateway maintains continuous links with LoRa edge nodes and the cloud backend to guarantee uninterrupted monitoring.
- *Static IP / Reserved DHCP:* A fixed IP configuration prevents reconnection failures due to IP changes after power cycles.

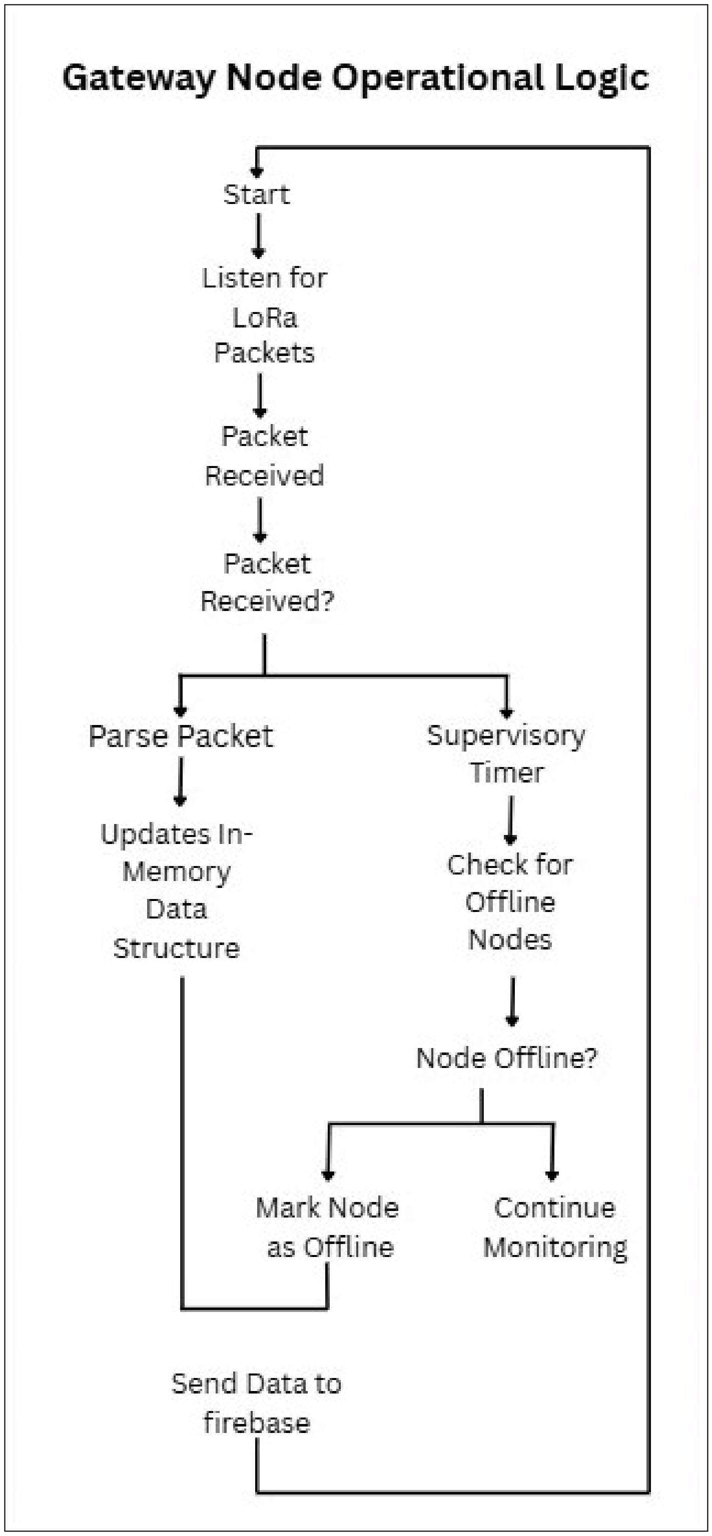

Fig. 3. Flowchart of the Gateway Node's operational logic, showing LoRa reception, state management, offline checking, and cloud synchronization.

- *mDNS Support:* Multicast DNS enables addressing via hostname (e.g., gateway.local), simplifying local communication and reducing manual configuration.
- *Watchdog Timer:* A watchdog mechanism automatically reboots the gateway upon software hangs, ensuring self-recovery.
- *Fail-Safe Operation:* In case of gateway failure, streetlights revert to a timer-based fallback mode to maintain illumination.

VI. EXPERIMENTAL SETUP AND RESULTS

To validate the system's design and performance, a functional prototype was developed and tested in a real-world environment simulating a small network of streetlights.

Fig. 4. Photograph of the prototype streetlight model.

A. Experimental Setup

The setup consisted of four transmitter nodes and one gateway node.

- **Transmitter Nodes:** Each node was housed in an IP67-rated weatherproof enclosure to protect against environmental factors like rain and dust, which is critical for outdoor deployments. The ambient sensor was positioned on top of the enclosure, facing the sky, with a small shield to block direct light from the lamp. The luminaire sensor was placed in a tube-like housing aimed directly at the lamp to isolate its reading. The nodes were mounted on poles at a height of 4 meters.
- **Gateway Node:** The gateway was placed in a central location with access to a Wi-Fi network.
- **Testing Environment:** The tests were conducted in a mixed residential and commercial area in Nagpur, with moderate building density and foliage, over a 48-hour period covering multiple day/night cycles.

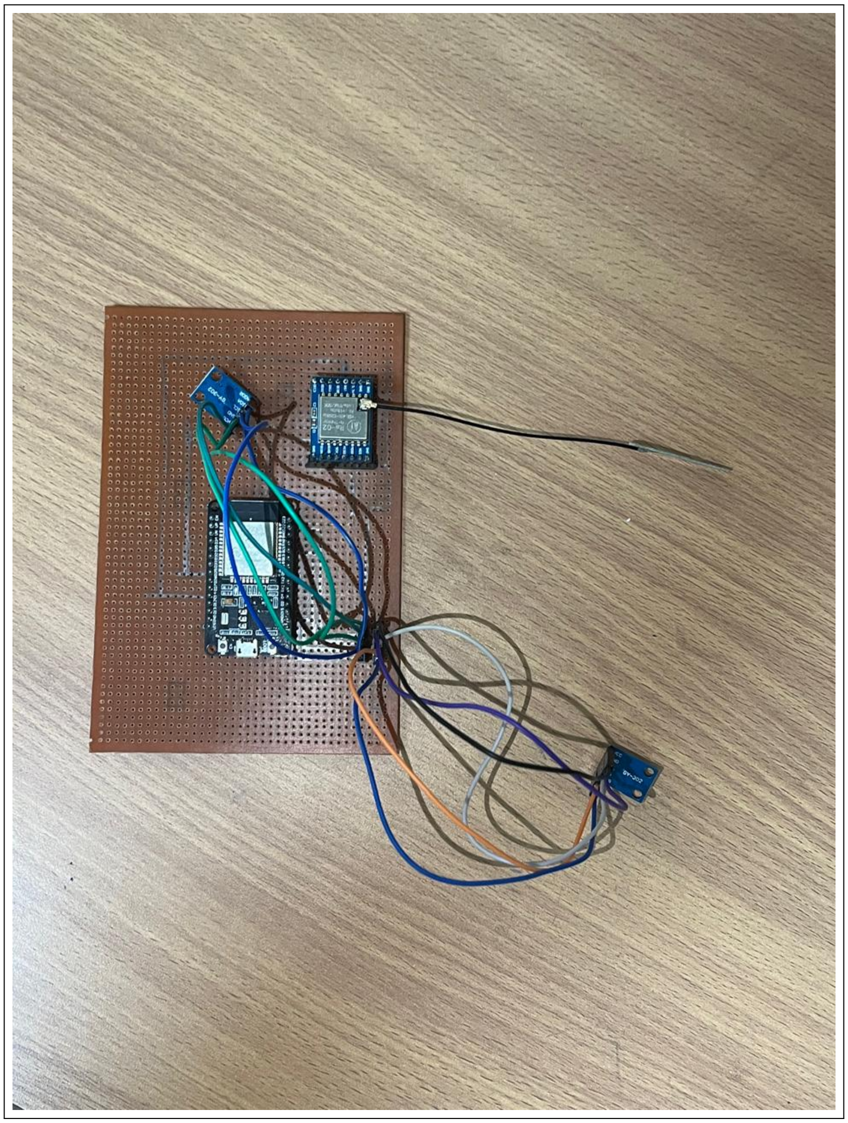

Fig. 5. Photograph of the prototype transmitter node's circuit board.

B. Results

The system's performance was evaluated based on three key metrics: LoRa communication range, end-to-end latency, and communication reliability.

- **Communication Range:** The LoRa communication proved to be robust. In the congested test area with obstacles, a reliable communication link was maintained up to a distance of 40-80 meters. In a line-of-sight test, the range extended to 120-180 meters. This confirms LoRa's suitability for reliable short-to-medium range communication in this application.
- **End-to-End Latency:** Latency was measured as the time from the transmitter node sending a LoRa packet to the corresponding status update appearing on the web dashboard. The average latency was measured to be 7.8 seconds. This rapid response time is more than sufficient for the non-time-critical application of fault monitoring.
- **Reliability:** Over the 48-hour test period, the four nodes transmitted approximately 2304 packets in total (assuming a 5-minute interval). The gateway successfully received and processed 2260 packets, resulting in a Packet Delivery Ratio (PDR) of 98.1%. The offline detection mechanism correctly identified the missed packets but did not trigger a false "Offline" alarm due to the 15-minute timeout window, demonstrating the system's resilience to occasional packet loss.

The results, summarized in Table III, confirm that the proposed system meets its design goals of being reliable, responsive, and suitable for long-range communication.

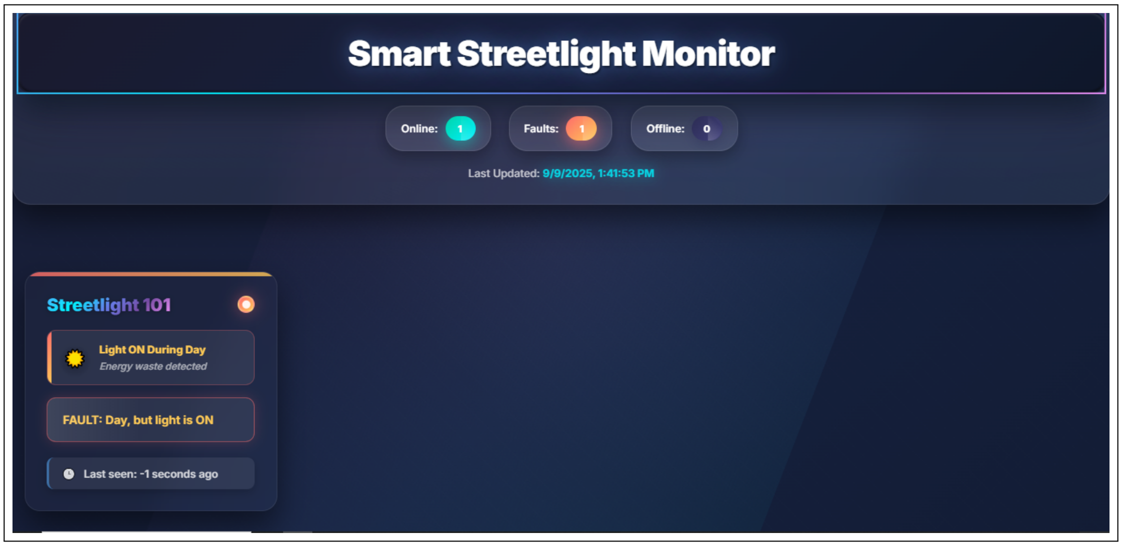

Fig. 6. Photograph of web dashboard

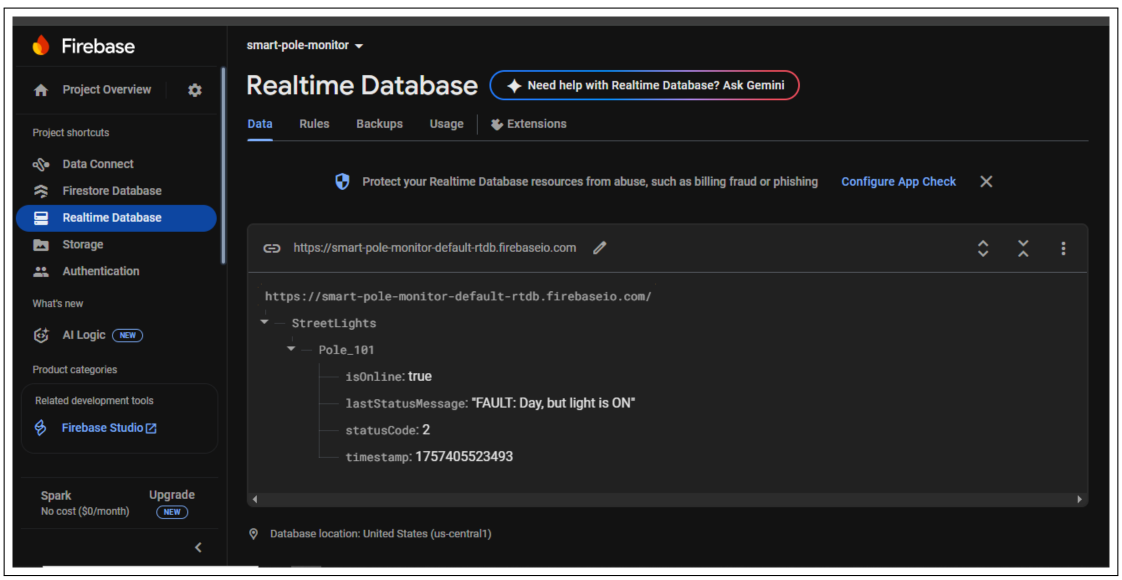

Fig. 7. Screenshot of the Firebase Realtime Database showing a detected fault condition for 'Pole_101'.

TABLE III
SUMMARY OF PERFORMANCE METRICS

Metric	**Value**
LoRa Range (Congested)	40-80 meters
LoRa Range (Line-of-Sight)	120-180 meters
Average Latency	7.8 seconds
Packet Delivery Ratio	98.1%

VII. DISCUSSION

The experimental results strongly validate the feasibility and effectiveness of the proposed decentralized streetlight monitoring system. The high packet delivery ratio of 98.1% in a real-world urban setting underscores the robustness of LoRa communication for this application. The occasional missed packet is effectively handled by the system's stateful design; since the dashboard displays the last known status, a single lost packet does not result in a loss of information, and the 15-minute timeout for offline detection prevents transient communication issues from generating false alarms.

The key innovation of this work, the decentralized dual-sensor fault detection algorithm, proved to be highly effective. It successfully identified and reported fault conditions (e.g., manually covering a lamp at night to simulate a failure) within seconds. This edge-level intelligence is a significant advantage over centralized systems, as it reduces the amount of data transmitted, lowers cloud computation costs, and increases the overall resilience of the network. The implication for a municipality is a shift from a reactive to a proactive maintenance model. A latency of under 10 seconds means a fault is known at the central office almost instantly, compared to the hours or days it might take for a manual patrol or citizen to report it. This rapid detection directly translates to improved public safety and quicker resolution of energy-wasting faults.

From a scalability perspective, the system is well-architected. A single gateway's ability to cover a radius of 40-80 meters in congested areas means it can effectively manage nodes within a localized cluster. To cover an entire city, one would simply need to deploy additional gateways, all of which would report to the same Firebase backend, creating a unified and infinitely scalable monitoring network.

While the prototype was successful, we acknowledge several areas for improvement in a production-grade deployment. The current LoRa communication is unencrypted, which poses a security risk. Implementing AES128 encryption on the data packets is a necessary next step. Furthermore, the firmware on the nodes is updated manually. An Over-the-Air (OTA) update mechanism would be essential for managing a large fleet of deployed devices. Finally, the prototype was powered via a DC adapter; a production version would require an integrated AC-DC power supply to draw power directly from the streetlight's main line, along with a small battery backup to report power failures.

VIII. CONCLUSION AND FUTURE WORK

This paper has presented the design, implementation, and evaluation of a low-cost, decentralized IoT system for real-time fault detection in urban streetlights. By leveraging autonomous edge computing with a novel dual-sensor algorithm, low-power LoRa communication, and a real-time cloud dashboard, the system provides an effective solution to the shortcomings of traditional streetlight management. The successful prototype validation demonstrates its high reliability, low latency, and excellent range, making it a practical and economically viable solution for municipalities seeking to build smarter, safer, and more efficient cities.

Future work will focus on enhancing the system for production deployment. Our immediate priorities are:

- **Security Enhancement:** Implementing end-to-end AES128 encryption for all LoRa transmissions. The ESP32 includes a hardware cryptographic accelerator, making it possible to implement strong encryption with minimal impact on performance and power consumption. A pre-shared key system would be employed for this private network.
- **Power System Integration:** Designing and integrating a robust AC-DC SMPS power supply with a rating of 5V/1A, featuring overvoltage and overcurrent protection. This will be coupled with a small Li-Po battery and charging circuit to act as an Uninterruptible Power Supply (UPS), allowing the node to transmit a "Mains Power Lost" message before shutting down.
- **Over-the-Air (OTA) Updates:** Developing a secure and reliable OTA update mechanism. For the gateway, this can be implemented over Wi-Fi. For the LoRa nodes, a more complex protocol will be developed where the gateway broadcasts firmware updates in small, checksum-verified chunks. This is a critical feature for long-term maintainability.
- **Advanced Analytics and Integration:** Enhancing the dashboard to include historical data analysis, trend visualization, and the automatic generation of maintenance reports. A future version could integrate with other municipal systems, for example, correlating streetlight outages with crime data or traffic flow information to provide deeper insights for city planners.

REFERENCES

[1] P. S. V. S. S. S. S. Sridhar, "Smart Street Lighting System for Urban Environments," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 6, pp. 2496–2500, 2019.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for Smart Cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[3] S. K. Sood, D. K. Sandhu, and K. Singla, "An IoT-based framework for smart street lighting system," in *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*. IEEE, 2017, pp. 196–200.

[4] M. Centenaro *et al.*, "LoRaWAN: Long-Range, Low-Power Internet of Things," *Future Internet*, 2016.

[5] A. Lavric, V. Popa, and I. L. Cretu, "A Power Line Communication Based System for Smart Street Lighting," in *2015 International Symposium on Signals, Circuits and Systems (ISSCS)*. IEEE, 2015, pp. 1–4.

[6] A. Anvari-Moghaddam, H. Monsef, and A. Rahimi-Kian, “Optimal smart home energy management considering energy saving and a comfortable lifestyle,” *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 324–332, 2015.
[7] M. T. Erdem, İ. Küçük, and M. Z. Kurt, “A smart street lighting system based on GPRS,” in *2016 International Conference on Engineering and Natural Sciences (ICENS)*, 2016, pp. 1045–1049.
[8] S. K. Singh, “Low Power Wide Area Networks (LPWAN) for IoT Applications,” *IoT Magazine*, 2018.
[9] O. I. Osin and K. A. Ghavami, “A Survey of LoRa Technology and its Applications,” in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE, 2021, pp. 1–6.
[10] L. Leonardi, L. Lo Bello, and G. Patti, “A LoRaWAN-Based Control and Monitoring System for Public Street Lighting,” in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2018, pp. 2382–2387.
[11] C. B. L. G. G. L. S. T. Callebaut, L. Van der Perre, and M. Weyn, “A Large-Scale Street Lighting Control System using LoRaWAN,” in *2019 Global IoT Summit (GIoTS)*. IEEE, 2019, pp. 1–6.
[12] L. Hou *et al.*, “Automated Streetlight Fault Diagnosis using LoRa,” *Sensors*, vol. 22, no. 9, 2022.
[13] S. Rajalakshmi *et al.*, “Fault Detection in IoT-Enabled Smart Lighting,” *IEEE IoT Journal*, 2022.
[14] L. Bellini, “Fault Tolerance in Wireless Sensor Networks for Urban Lighting,” *Sensors*, 2020.
[15] Espressif Systems, *ESP32 Technical Reference Manual*, 2022.
[16] Semtech Corp., “SX1278 LoRa Transceiver Datasheet,” 2021.
[17] Google, “Firebase Realtime Database Documentation,” [Online]. Available: https://firebase.google.com/docs/database, 2025, accessed: Sep. 4, 2025.
[18] M. H. H. S. K. V. S. Sridhar, “A LoRa based smart street light management system,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 10587–10599, 2021.
[19] S. Khemakhem and L. Krichen, ”A comprehensive survey on an IoTbased smart public street lighting infrastructure,” *Franklin Open*, August 2024, pp. 1-15.
[20] G. S. Rani, G. N. Kumar, I. Anusha, M. V. Jyothi, and P. N. Kumar, ”IoT Based Street Light Controller and Monitoring system,” *SSRN Electronic Journal*, February 26, 2024.
[21] A. K. Inde and A. S. Gundale, ”LoRa-based Monitoring of Street Lights: A New Approach for Managing Street Lights,” *Grenze International Journal of Engineering and Technology (GIJET)*, vol. 10, no. 2, 2024.
[22] J. Mathew, R. Rajan, and R. Varghese, ”IOT BASED STREET LIGHT MONITORING & CONTROL WITH LoRa/LoRaWAN NETWORK,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 11, November 2019.
[23] B. Vidhate, S. Patil, and S. Somani, ”LORA BASED STREET LIGHT AUTOMATION FOR SMART-CITIES,” *International Journal of Electrical Engineering & Technology (IJEET)*, vol. 9, no. 5, pp. 64-72, September-October 2018.
[24] Prashanth. S U, Rakshanraj. J, and A. M. C. Blessy, ”IoT Based Smart Street Lighting System,” in *ETEDM 2022 Conference Proceedings*, July 29, 2022.
[25] M. Saifuzzaman, N. N. Moon, and F. N. Nur, ”IoT based street lighting and traffic management system,” in *2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC)*, 2017, pp. 121-124.
[26] S. Sangameshwar, A. Sagi, H. Channapragada, N. Preethi, and V. V. Mounika, ”Nano Uno-ML-IoT Based Smart Street Light System with Centralized Monitoring and Control,” *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, December 24, 2023.
[27] R. P. S. Kumar and R. Vadivel, ”IOT based smart street light fault detection management system,” *World Journal of Advanced Research and Reviews*, vol. 21, no. 3, pp. 1659-1666, 2024.
[28] H. Nurwarsito and R. D. Prasetyo, ”Implementation of Lora on the Wireless Sensor Network-based Public Street Lighting Monitoring System,” *International Journal of Advanced Computer Science and Engineering (IJACEN)*.
[29] S. K. Pappula, L. L. Eemani, and P. S. Rama, ”IoT based Street Light Auto Intensity Control and Fault Detection,” *International Journal of Engineering Research and Technology (IJERT)*, May 22, 2022.
[30] Allegro MicroSystems, ”CT100 1D Linear Sensor,” Datasheet Rev. 7, May 15, 2025.
[31] Allegro MicroSystems, ”Non-Intrusive Current-Sensing Using CT100 TMR Linear Sensor,” Application Note AN118, Rev. 1, 2019.
[32] Avnet Abacus/Murata, ”Murata CT100 Series TMR Sensors,” Product Information, January 22, 2021.
[33] OpenEnergyMonitor, ”CT sensors - An Introduction,” Technical Guide.
[34] C. Kancharapu, Vamsi. L, Sai. K, Sai Teja. K, and Bharath Kumar. N, ”Smart Fault Detection and Maintenance for IoT Based Street Lighting Systems,” *Science, Technology and Development*, vol. 13, no. 3, pp. 161-167, March 2024.
[35] A. K. Inde and A. S. Gundale, ”LoRa-based Monitoring of Street Lights: A New Approach for Managing Street Lights,” *Grenze International Journal of Engineering and Technology*, vol. 10, no. 2, pp. 5009-5015, June 2024.
[36] P. P. M. Krishna *et al.*, ”Automatic Street Light System,” *Journal of Computational Analysis and Applications*, vol. 33, no. 8, pp. 3710-3716, 2024.
[37] Nishanti Naidu *et al.*, ”Smart Street Lighting Enabled By Wireless Sensor Networks: A Path Towards Sustainable Urban Infrastructure,” *International Journal of Wireless Sensor Networks*, February 2025.
[38] M. Sudharsh, ”Street Light Fault Detection and Control System Using ESP32 and IoT,” GitHub Repository, July 2024.
[39] A. Sravani *et al.*, ”Street Light Fault Detection and Control System Urban Glow,” *International Journal of Engineering Research in Electronic and Communication Engineering*, July 2024.