JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

REMOTE ENVIRONMENTAL MONITORING AND ALERT SYSTEM

[1] D.Arun Chowdary, [2] Madham Purushotham, [3] Ms. Preetha Kamath B
[1-2]UG Student, [3]Assistant Professor
[1-3] Electronics and Communication Engineering,
[1-3] K S Institute of Technology, Bangalore, India

Abstract: This project presents an IoT-based environmental monitoring and alert system that continuously measures key environmental parameters such as temperature, humidity, gas levels, and light intensity. The system is built using an ESP32 microcontroller interfaced with DHT11, LDR, and gas sensors, along with a GPS module for location tracking. Data from the sensors is displayed in real-time on an LCD and simultaneously transmitted to the cloud platform, ThingSpeak, enabling remote monitoring and historical data logging. The GPS integration allows precise tracking of environmental conditions at specific locations, which is useful for outdoor monitoring and data-driven decision-making.

The system also features an alert mechanism that notifies users via a mobile application whenever sensor readings exceed predefined thresholds, enhancing safety and responsiveness. By combining IoT technology with mobile accessibility, this project enables real-time environmental monitoring and automated alerts, making it suitable for applications such as air quality tracking, smart cities, and safety monitoring in industrial or residential areas. The modular design ensures that additional sensors or features can be easily integrated, off,ering a flexible and scalable solution for environmental data collection and management.

Index terms : GPS module, IoT-based environmental monitoring and alert system, ESP32 microcontroller, data logging

I.INTRODUCTION

Environmental monitoring is crucial for understanding and managing the quality of air, water, and surrounding conditions, especially in urban and industrial areas. Rapid industrialization, urbanization, and vehicular emissions have increased environmental hazards, affecting human health and safety. Traditional environmental monitoring methods often involve manual data collection, which is time-consuming, labor-intensive, and prone to delays in detecting critical changes.Advances in IoT technology have enabled real-time monitoring of environmental parameters using sensor networks connected to cloud platforms. By integrating sensors, microcontrollers, and communication modules, IoT systems can collect, process, and transmit data continuously. In this project, an IoT-based environmental monitoring system is developed using ESP32, DHT11 (temperature and humidity), LDR (light intensity), and a gas sensor, along with GPS for location tracking. The system logs environmental data in real-time to ThingSpeak and provides alerts via a mobile application when abnormal conditions are detected.

II. LITERATURE SURVEY

1. This paper presents a low-cost IoT-based environmental monitoring system using Arduino UNO, DHT11 sensor, and an ESP8266 Wi-Fi module. The system monitors temperature and humidity and uploads the data to the ThingSpeak cloud platform. An Android application retrieves the data through REST APIs, enabling real-time environmental monitoring. The work demonstrates the effectiveness of affordable hardware and cloud services.[1]
2. This work proposes an environmental monitoring system based on a wireless sensor network integrated with IoT technology. It measures temperature, humidity, and carbon monoxide (CO) levels. The system is

modular and adaptable, allowing deployment in different environmental conditions. Sensor data is stored in the cloud, supporting scalability and remote access.[2]

3. This paper describes an IoT-based system using an Arduino board, GPRS communication, and sensors for temperature, humidity, and gas detection. The collected data is transmitted to the cloud platform and visualized through a remote dashboard. The study highlights the feasibility of low-cost IoT solutions for both indoor and outdoor environmental monitoring.[3]

4. This work focuses on air quality monitoring using IoT sensor nodes capable of measuring pollutants such as carbon monoxide (CO), ozone (O_3), and particulate matter (PM). The system provides real-time and historical data visualization, enabling effective pollution tracking and environmental analysis.[4]

5. This paper addresses the issue of inaccuracy in low-cost air pollution sensors. A spatial-temporal graph attention network is used to calibrate sensor data by fusing information from multiple sensor nodes. The proposed approach significantly improves data accuracy and reliability in IoT-based air quality monitoring systems.[5]

6. This study develops an IoT-based system to monitor temperature, humidity, and air pollution parameters. Sensors are connected to a microcontroller, and data is uploaded to a cloud server. The system supports real-time monitoring and web-based visualization, demonstrating continuous environmental observation.[6]

7. This paper presents an environmental monitoring system implemented using a Raspberry Pi as the processing unit. The system monitors air quality and temperature and transmits data to the cloud. The study highlights the applicability of IoT in smart city and urban pollution monitoring scenarios.[7]

8. This work proposes a smart environmental monitoring system using IoT and wireless communication technologies. The system enables real-time alerts, cloud-based data access, and remote monitoring. Emphasis is placed on low-cost deployment and scalability, making it suitable for large-scale applications.[8]

9. This paper describes an IoT-based system designed to monitor air pollution levels using gas sensors. The system enables remote access and generates alerts when pollution levels exceed predefined thresholds. The study demonstrates the usefulness of IoT in urban environmental management.[9]

10. This work focuses on IoT-based sensing and automation within smart home environments. Although the application domain differs, the concepts of real-time monitoring, data communication, and automated response mechanisms are applicable to environmental monitoring systems.[10]

11. This paper presents an IoT-based system for continuous weather monitoring, focusing on temperature and humidity measurement. Sensor data is stored in the cloud, allowing remote access and visualization. The study demonstrates the effectiveness of IoT for real-time weather monitoring.[11]

12. This work focuses on monitoring harmful air pollutants using IoT-enabled sensors. The system provides real-time alerts and supports historical data analysis. The study highlights the importance of IoT-based pollution monitoring for public health and environmental safety.[12]

13. This paper proposes an environmental monitoring system using wireless sensor networks integrated with IoT platforms. The system is designed to be scalable, energy-efficient, and reliable, making it suitable for distributed and large-area monitoring applications.[13]

14. This study presents an IoT-based air quality monitoring system designed for smart city environments. It emphasizes data analytics, visualization, and decision-support mechanisms. The work demonstrates how IoT can assist authorities in pollution control and urban planning.[14]

15. This paper provides a comprehensive overview of IoT architectures, enabling technologies, communication protocols, and applications. Environmental monitoring is identified as a major application area. The study offers a strong theoretical foundation for the development of IoT-based environmental monitoring systems.[15]

III. PROBLEM STATEMENT

1. Traditional environmental monitoring methods rely on manual data collection, periodic sampling, and laboratory analysis, which are often slow, costly, and limited in spatial and temporal coverage. As a result, environmental hazards such as air pollution spikes, water contamination, forest fires, or abrupt weather changes often go undetected until the damage has already occurred. Existing monitoring infrastructures are typically sparse, inflexible, and unable to provide real-time alerts necessary for rapid response and effective mitigation.

2. Furthermore, remote and rural areas lack stable monitoring networks due to the absence of power supply, reliable connectivity, and high installation and maintenance costs. Low-cost sensor technologies, though widely available, face challenges such as calibration drift, limited accuracy, and data reliability issues. Additionally, collected data from heterogeneous systems often lack standardization, making integration, interpretation, and decision-making difficult.
3. There is also a growing need for automated, scalable, and intelligent systems that not only collect environmental data but can also analyze trends, detect anomalies, and provide early warnings with minimal human intervention. Security and privacy concerns in IoT-based systems further complicate system deployment and long-term operation.
4. Therefore, there is a significant need to develop a smarter, low-power, automated, and reliable Remote Environmental Monitoring and Alert System capable of continuous sensing, secure data communication, intelligent processing, and timely notification to prevent environmental degradation and support disaster preparedness.

IV. OBJECTIVES

The specific goals of this project are:
1. To design and implement a real-time environmental monitoring system using ESP32 and multiple sensors (DHT11, LDR, and gas sensor).
2. To integrate GPS for location-based logging of environmental data.
3. To transmit sensor data to the cloud platform, ThingSpeak, for remote monitoring and historical analysis.
4. To provide real-time alerts via a mobile application when environmental parameters exceed predefined safe limits.
5. To create a user-friendly interface for local display of sensor readings on an LCD and remote monitoring on mobile devices.
6. To develop a scalable and modular system that can accommodate additional sensors and features in the future.

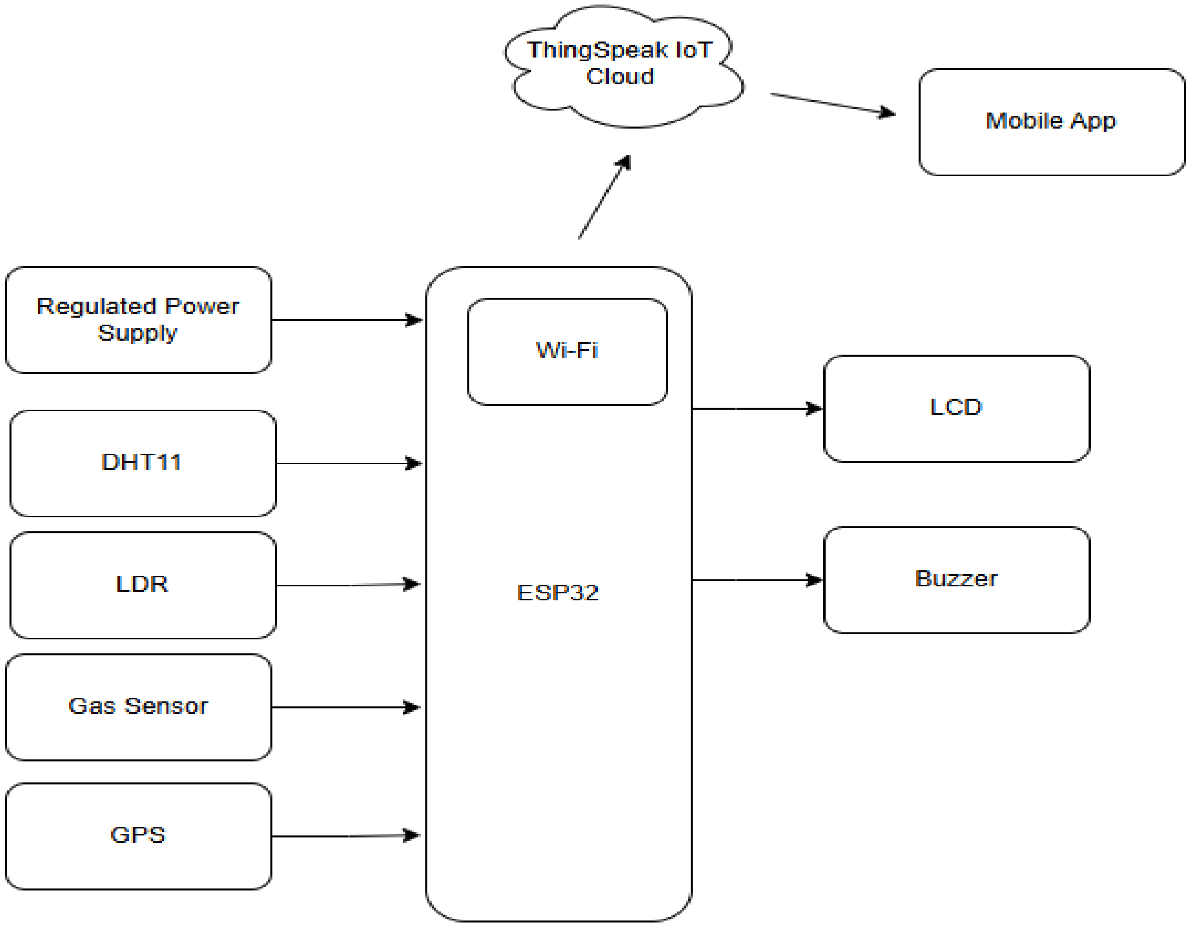
V. METHODOLOGY

The methodology for developing a Remote Environmental Monitoring and Alert System involves several sequential stages, from sensing to alert generation and performance evaluation. The steps are outlined below:

1.Requirements Analysis and Parameter Selection:
Identify the target environmental variables based on the application (e.g., air quality, water quality, soil moisture, weather). This includes defining acceptable threshold values, regulatory standards, and the desired spatial and temporal monitoring frequency.
2.Sensor and Hardware Selection
3.Select appropriate environmental sensors such as gas sensors (CO_2, CO, NO_2), particulate sensors (PM2.5/PM10), temperature and humidity sensors, water pH/turbidity sensors, and fire or flood detectors.
4.A suitable microcontroller or embedded platform (e.g., ESP32, Raspberry Pi, Arduino) is chosen to interface with sensors. Power supply solutions (battery, solar panel, DC adaptor) are determined based on deployment location.
5.Data Acquisition and Preprocessing
6.Sensor data is periodically collected and filtered to remove noise or invalid readings. Preprocessing techniques such as calibration, normalization, smoothing, and outlier detection are applied to improve data accuracy and reliability before transmission.
7.Communication and Networking
8.A suitable communication protocol is implemented based on range and power requirements.
Possible options include:
- Short-range: Wi-Fi, Bluetooth Low Energy (BLE)
- Long-range / Remote: LoRaWAN, NB-IoT, GSM/4G/5G
- MQTT, HTTP, or CoAP protocols are used for transmitting data securely from edge devices to a remote server or cloud platform.
- Cloud Storage and Data Management

- Collected sensor data is stored in a cloud database or edge-storage buffer for processing and visualization. Time-series databases, IoT platforms, or Firebase/Thingspeak/Node-RED can be used for long-term storage and retrieval.
- Data Processing and Decision Logic
- Algorithms are applied to analyze data trends, detect abnormalities, or predict hazardous conditions. Two common approaches are used:
- Threshold-based alerts (fixed environmental limits)
- Machine Learning / AI models for anomaly detection and forecasting (optional, depending on system complexity)
- Alert Generation and Notification
- If the system detects conditions exceeding predefined limits, automated alerts are generated and delivered to users through:
- Mobile app notifications
- SMS/Email alerts
- Dashboard warnings
- Sirens or automated actuator triggers (e.g., turning on pumps, alarms)

VI. BLOCK DIAGRAM

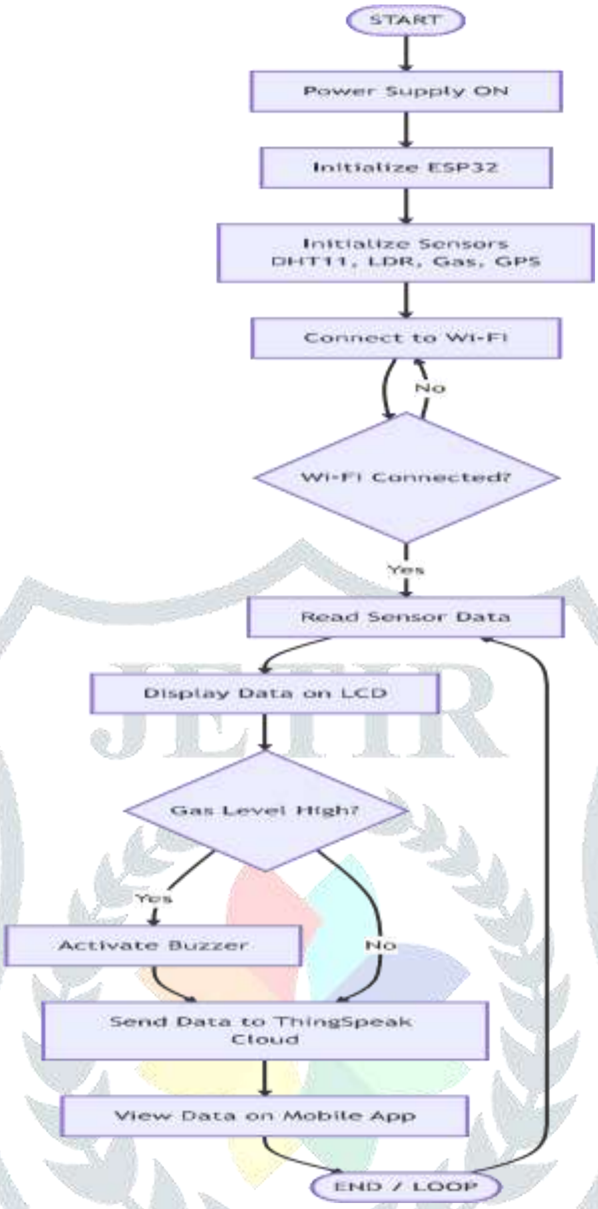


The block diagram represents a Remote Environmental Monitoring and Alert System. The system uses different sensors such as a DHT11 (for temperature and humidity), an LDR (for light intensity), a gas sensor (for detecting harmful gases), and a GPS module (for location tracking). All these sensors send their readings to the ESP32 microcontroller.

The ESP32 processes the collected data and then displays the values on an LCD screen for local monitoring. If any reading goes beyond the safe limit, the buzzer will turn on to give an immediate alert.

Using its built-in Wi-Fi, the ESP32 also uploads the sensor data to the ThingSpeak IoT cloud. Through a mobile app, users can view the real-time environmental data from anywhere. A regulated power supply ensures the system operates smoothly.

VII. FLOW CHART



The flowchart illustrates the operational process of the Remote Environmental Monitoring and Alert System. The system begins with the Start step, where the device is powered on. After starting, the system moves to the Initialization stage, where essential components such as Wi-Fi connectivity, sensors, and cloud services are configured and made ready for operation.

1. Start:
The system begins its operation when power is supplied or the device is turned on.

2. Initialize System:
In this step, all required components such as Wi-Fi, sensors, and cloud connectivity are set up and initialized. This ensures the system is ready to collect and transmit data.

3. Read Sensors:
The controller collects real-time environmental readings from connected sensors like the DHT11 (temperature and humidity), LDR (light intensity), gas sensor, and GPS module.

4. Check Threshold:
The collected sensor values are compared with predefined safe limits. This decision step determines whether the environment is normal or if a critical condition is detected.

5. Send Alert (If Required):
If any parameter exceeds the threshold value, the system triggers an alert. This alert may be sent through a buzzer, mobile notification, or application warning.

6. Log Data (Sensor + GPS):
All sensor readings, along with location data from the GPS module, are stored for record-keeping and future analysis.

7. Transmit to ThingSpeak:
The logged data is sent to the ThingSpeak cloud server for storage, processing, and visualization.

8. Display on Dashboard:
The transmitted data becomes visible on a mobile or web dashboard, allowing users to remotely monitor live environmental conditions.

9. Wait/Delay and Repeat:
The system waits for a defined time interval before the next monitoring cycle begins, ensuring continuous and automatic operation.

VIII. RESULT

The developed IoT-based environmental monitoring system successfully demonstrated real-time measurement, logging, and alerting for temperature, humidity, light intensity, and gas levels. Key observations include:

Sensor Accuracy: The DHT11, LDR, and gas sensors provided consistent readings with minor deviations, suitable for environmental monitoring purposes.

Real-Time Data Logging: Data was successfully uploaded to ThingSpeak, enabling remote monitoring and historical data analysis.

Mobile App Monitoring: Users could access environmental data via a mobile app, with real-time updates from the ESP32 module.

Alert System: Alerts were triggered accurately when environmental parameters exceeded predefined thresholds, allowing timely user intervention.

GPS Integration: The system successfully logged the location of measurements, making it suitable for outdoor and multi-location environmental monitoring.

Overall, the system proved to be reliable, cost-effective, and efficient for monitoring environmental parameters in both indoor and outdoor settings.

IX. APPLICATION

1. Air Quality Monitoring
2. Tracking pollutants and gas leaks in urban areas, factories, and industrial zones.
3. Smart Cities
4. Monitoring environmental conditions such as temperature, humidity, and light levels to improve urban living and infrastructure planning.
5. Industrial Safety
6. Detecting gas leaks, smoke, or extreme environmental conditions in factories and warehouses.
7. Agriculture & Greenhouses
8. Monitoring temperature, humidity, and light for better crop growth and automated irrigation control.
9. Residential & Office Safety
10. Ensuring safe indoor air quality and alerting users in case of hazardous gas accumulation.
11. Research & Environmental Studies
12. Collecting location-specific environmental data for studies on climate, pollution, or energy efficiency

X. FUTURE SCOPE:

Although the developed IoT-based Environmental Monitoring and Alert System performs effectively, several enhancements can be implemented to improve functionality, accuracy, and scalability in future work:

1. InegrationofAdditionalSensors:

The system can be expanded to include sensors for air quality indices (PM2.5, PM10), carbon dioxide (CO_2), sound levels, soil moisture, and water quality parameters to enable more comprehensive environmental monitoring.

2. ImprovedSensorAccuracy:

Higher-precision sensors can replace existing modules to enhance measurement accuracy and reliability, particularly for industrial or research-grade applications.

3. EdgeComputingImplementation:

Processing data locally on the ESP32 or edge devices can reduce latency, minimize cloud dependency, and improve real-time decision-making.

4. EnergyOptimizationandPowerManagement:

Integrating solar power and low-power communication protocols can improve energy efficiency, making the system suitable for long-term outdoor deployments.

5. .SystemScalability:

The system can be expanded into a wireless sensor network, enabling large-scale environmental monitoring across multiple locations.

6. .EdgeComputingImplementation:

Processing data at the edge using the ESP32 can reduce latency, minimize cloud dependency, and enable faster real-time responses.

7. BigDataandLong-TermAnalytics:

Large-scale data collection over time can be leveraged for climate studies, pollution trend analysis, and research applications.

8. .AutomatedControlActions

The system can be extended to trigger automated responses such as activating ventilation systems, alarms, or sprinklers when hazardous conditions are detected.

XI. ACKNOWLEDGEMENT

The successful project is culmination of efforts of many people who have rendered their unconditional support. We would be dishonest without acknowledging these people.

Dr. Dilip Kumar K, Principal / Director, K. S. Institute of Technology, who has been a continuous source of inspiration to us.We are indebted to him for his encouragement in making the project a success.

Dr. P. N. Sudha, Dean Academics and Head of Department, Electronics and Communication Engineering, K. S. Institute of Technology, who has continuously encouraged, motivated and provided us valuable suggestions.

We are also thankful to our internal guide, Ms. Preetha Kamath B, Assistant Professor, Department of Electronics & Communication Engineering, K. S. Institute of Technology, for providing valuable support and guidance to make this project a success.

We are thankful to our project Coordinators Mrs. Bhargavi Ananth, Assistant Professor & Mrs. Divya D, Assistant Professor, Department of Electronics and Communication Engineering, K. S. Institute of

Technology, for their coordination and help.

Last but not the least the project would not have been a success without the grace of God and support of our Parents, Teaching faculty, non-teaching faculty and Friends.

XII.REFERENCES:

1. S. Zafar, G. Miraj, R. Baloch, D. Murtaza, K. Arshad, *"An IoT Based Real-Time Environmental Monitoring System"*, International Journal of Emerging Technology and Advanced Engineering, 2015.
2. Suhas Pandurang Nikam & Dr. S. M. Kulkarni, *"IoT Based Environmental Monitoring System"*, IJERT, 2018.
3. T. Jayasree & Dr. S. Rajasekaran, *"IoT-Based Environmental Monitoring System Using Arduino and GPRS"*, IJSRSET, 2019.
4. Prince Kazembe & Mtende Mkandawire, *"IoT-Based Air Quality Monitoring System"*, i-manager's Journal on Information Technology, 2020.
5. Keivan Faghih Niresi, Mengjie Zhao, Hugo Bissig, Henri Baumann & Olga Fink, *"Spatial-Temporal Graph Attention Fuser for Calibration in IoT Air Pollution Monitoring Systems"*, arXiv, 2023.
6. Espressif Systems, *ESP32 Technical Reference Manual*, Espressif, 2021.
7. ThingSpeak Documentation, MathWorks, 2022.
8. A.Kumar, N. Kumar, and R. Kumar, "Smart environmental monitoring system using IoT and cloud computing," *International Journal of Computer Applications*, vol. 174, no. 4, pp. 15–20, 2017.
9. M. A. Rahman, M. M. Islam, and M. S. Hossain, "IoT-based real-time monitoring system for environmental parameters," in *Proc. IEEE Int. Conf. on Electrical, Computer and Communication Engineering (ECCE)*, 2019, pp. 1–6.
10. S. K. Vishwakarma, P. Upadhyaya, B. Kumari, and A. K. Mishra, "Smart energy efficient home automation system using IoT," *International Journal of Advanced Research in Computer Science*, vol. 10, no. 1, pp. 45–50, 2019.
11. J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sept. 2013.
12. L. Da Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.
13. K. Ashton, "That 'Internet of Things' thing," *RFID Journal*, vol. 22, no. 7, pp. 97–114, 2009.
14. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.
15. N. D. Han and H. T. Hieu, "IoT-based environmental monitoring system using wireless sensor networks," *International Journal of Engineering Research and Technology*, vol. 8, no. 6, pp. 321–326, 2019.
16. R. K. Kodali and S. Yerroju, "Energy efficient smart home automation using IoT," in *Proc. IEEE Int. Conf. on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 1–5.
17. S. Li, L. Da Xu, and S. Zhao, "The Internet of Things: A survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
18. P. Sethi and S. R. Sarangi, "Internet of Things: Architectures, protocols, and applications," *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–25, 2017.
19. H. Boyes, B. Hallaq, J. Cunningham, and T. Watson, "The industrial Internet of Things (IIoT): An analysis framework," *Computers in Industry*, vol. 101, pp. 1–12, 2018.
20. S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A literature review," *Journal of Computer and Communications*, vol. 3, no. 5, pp. 164–173, 2015.