INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 4, April 2026 | ISSN: 2456-4184 | IJNRD.ORG

Design and Implementation of an IoT-Based Smart Home Automation Switch Using ESP8266

Prof. Harna M. BodeleGaurav S. Deshbhratar
Dheeraj U. Dhengre
Rohit R. Fulzele
Chaitanya S. Bawane
Department of Electronics & Tele - Communication Engineering, JDCOEM, Nagpur.

Abstract

This paper presents the design, implementation, and empirical evaluation of a low-cost IoT-based smart home automation switch using the ESP8266 microcontroller. The proposed system integrates Wi-Fi-based communication, relay-driven load control, and a mobile application interface utilizing MQTT and cloud-based services for real-time monitoring and remote operation.

Experimental validation was conducted over a 7-day period under realistic residential conditions. Results demonstrate a mean response latency of 278 ms for local MQTT communication, a command execution reliability of 98.5%, and a 14.6% reduction in energy consumption compared to conventional manual switching. The system architecture supports scalable multi-device integration while maintaining low hardware cost (₹450–600).

Despite its effectiveness, the system remains dependent on network availability and currently lacks end-to-end secure communication, which are identified as areas for future improvement. Overall, the proposed solution provides a cost-efficient and practical approach to smart home automation with quantified performance benchmarking.

Keywords:
IoT, Smart Home Automation, ESP8266, Wi-Fi Control, Relay Module, Mobile Application, Embedded Systems

I. INTRODUCTION

The rapid advancement of the Internet of Things (IoT) has enabled the transformation of conventional residential environments into intelligent, interconnected systems capable of real-time monitoring and control. Smart home automation has emerged as a key application domain, allowing users to remotely manage household appliances, improve energy efficiency, and enhance overall convenience through internet-enabled devices. Low-cost Wi-Fi microcontrollers, particularly the ESP8266, have significantly accelerated the development of such systems by providing integrated connectivity, ease of programming, and broad ecosystem support.

Despite these developments, a large proportion of residential infrastructure continues to rely on traditional electrical switching mechanisms that lack remote accessibility, real-time feedback, and energy optimization capabilities. Existing commercial smart home solutions often suffer from high deployment costs, complex installation requirements, and dependence on proprietary cloud ecosystems, limiting their accessibility and flexibility. Furthermore, many reported implementations in the literature do not provide comprehensive experimental validation, particularly in terms of latency, reliability, and energy efficiency under real-world operating conditions.

To address these limitations, this work presents the design and implementation of a cost-effective IoT-based smart home automation switch using the ESP8266 microcontroller. The proposed system integrates relay-based load control with Wi-Fi communication and a mobile application interface, enabling both local and remote operation through MQTT and cloud-based services. In contrast to several existing solutions, this study emphasizes empirical performance evaluation, including quantitative analysis of response time, command reliability, and energy consumption over an extended testing period.

The key contributions of this paper are as follows:

- Design and implementation of a low-cost (₹450–600) smart switching system using the ESP8266 platform
- Development of a dual-mode control mechanism supporting both local (MQTT) and remote (cloud-based) operation
- Comprehensive experimental evaluation of system performance, including latency, reliability, and energy efficiency
- Comparative analysis with existing ESP8266-based home automation systems from the literature

The remainder of this paper is organized as follows. Section II reviews related work and identifies research gaps. Section

III describes the system architecture and methodology. Section IV presents the experimental results and performance analysis. Section V concludes the paper and outlines directions for future work.

II. Literature Survey

A. Overview

The rapid advancement of the Internet of Things (IoT) has enabled the development of intelligent residential environments in which interconnected devices facilitate automation, remote monitoring, and energy management. Smart home automation systems have gained considerable attention due to their ability to improve user convenience, operational efficiency, and energy utilization. Recent research has primarily focused on low-cost implementations using Wi-Fi-enabled microcontrollers such as the ESP8266 and ESP32, which provide integrated connectivity, ease of programming, and scalability.

Existing work in this domain can be broadly classified based on system architecture, communication protocols, and functional capabilities. This section presents a critical analysis

of representative studies, highlighting their design approaches, strengths, and limitations.

B. Architectural Approaches

Smart home automation systems are typically implemented using centralized, distributed, or hybrid architectures.

Centralized architectures, as proposed by C. Stolojescu-Crisan et al., rely on a gateway or cloud server for device coordination and control. While such systems simplify management, they introduce a single point of failure and increased latency due to cloud dependency.

Distributed architectures, such as the approach presented by M.
S. Munoz-Abad et al., utilize multiple interconnected nodes that operate collaboratively. These systems improve fault tolerance and scalability but increase synchronization complexity and system overhead.

Hybrid architectures combine local processing with cloud-based services to balance responsiveness and accessibility. However, many existing hybrid systems still exhibit excessive reliance on cloud infrastructure, limiting system autonomy during network disruptions.

C. Communication Protocols

Communication protocols significantly influence system performance, responsiveness, and scalability.

HTTP-based systems are widely used due to their simplicity and compatibility with web technologies. However, their request–response model introduces higher latency and communication overhead, making them less suitable for real-time control applications.

In contrast, Message Queuing Telemetry Transport (MQTT) has emerged as a preferred protocol for IoT systems due to its lightweight publish–subscribe architecture. Studies such as those by P. Arora demonstrate that MQTT reduces bandwidth consumption and enables faster, asynchronous communication compared to HTTP.

Hybrid communication approaches combining Wi-Fi with cellular technologies (e.g., GSM) have also been proposed to enhance system availability during network failures. While these approaches improve reliability, they increase system cost and design complexity, limiting their practicality for low-cost deployments.

D. Functional Capabilities and Performance Evaluation

Several studies have explored functional enhancements such as remote monitoring, scheduling, and energy management in smart home systems. ESP8266-based implementations integrated with cloud platforms provide convenient remote access; however, many lack comprehensive experimental validation.

A critical limitation across existing literature is the absence of standardized performance benchmarking. Metrics such as response latency, command reliability, and energy efficiency are often either partially reported or not evaluated under realistic conditions. Additionally, most implementations rely on short-duration testing or simulation-based analysis, which does not accurately represent real-world deployment scenarios.

Energy optimization is frequently discussed but rarely quantified using detailed measurement methodologies. Similarly, system scalability and multi-device performance are
often assumed rather than empirically validated.

Security Considerations

Security remains a major challenge in IoT-based smart home systems. Many existing implementations lack robust authentication and encryption mechanisms, exposing systems to potential cyber threats such as unauthorized access, data interception, and replay attacks.

Research by Y. Yang et al. highlights vulnerabilities in IoT communication frameworks, particularly in systems that do not implement Transport Layer Security (TLS) or secure key management practices. Although lightweight authentication protocols have been proposed in recent studies, their adoption in low-cost IoT systems remains limited due to computational constraints and implementation complexity.

E. Research Gaps

Based on the above analysis, several critical gaps are identified in existing smart home automation systems:

- Lack of comprehensive quantitative performance benchmarking, particularly for latency and reliability
- Excessive dependence on cloud infrastructure, reducing system autonomy
- Limited implementation of secure communication protocols, such as TLS-enabled MQTT
- Insufficient validation under real-world, long-duration operating conditions
- Minimal focus on cost-performance optimization for large-scale deployment
- Inadequate evaluation of system scalability and multi-device integration

F. Positioning of the Proposed Work

To address the identified limitations, the present work focuses on the design and empirical evaluation of a low-cost ESP8266-based smart home automation system with an emphasis on real-world performance benchmarking. The proposed system incorporates a dual-mode communication architecture supporting both local and cloud-based operation, thereby reducing latency and improving reliability.

Furthermore, this study provides detailed quantitative analysis of response time, command success rate, and energy consumption over an extended testing period. By integrating practical implementation with rigorous evaluation, the work contributes toward establishing more reliable and cost-effective smart home automation solutions.

III. Methodology and System Architecture

A. System Overview

The proposed system is designed as a low-cost, Wi-Fi-enabled smart switching solution that enables real-time monitoring and control of electrical appliances. The architecture follows a hybrid communication model, supporting both local network operation and cloud-based remote access. This dual-mode design reduces latency during local operation while maintaining accessibility over the internet.

The system consists of three primary layers: (i) the hardware layer comprising the ESP8266 microcontroller and relay interface, (ii) the communication layer utilizing MQTT/HTTP protocols over Wi-Fi, and (iii) the application layer represented by a mobile interface for user interaction. Commands generated by the user are transmitted through the communication layer, processed by the ESP8266 firmware, and executed via the relay module, with real-time feedback provided to the user interface.

B. System Flowchart:

Flowchart illustrates the operational flow of the proposed smart home automation system. The process begins with user interaction through the mobile application, where a control command (ON/OFF or scheduled task) is generated. The command is transmitted via either local Wi-Fi or cloud-based communication depending on network availability.

Upon reception, the ESP8266 validates the command and determines the corresponding control action. The firmware updates the appropriate GPIO pin, which triggers the relay module to actuate the connected appliance. A feedback signal is then transmitted back to the mobile application to ensure real-time synchronization. In case of connectivity failure, the watchdog mechanism initiates automatic reconnection, improving system reliability.

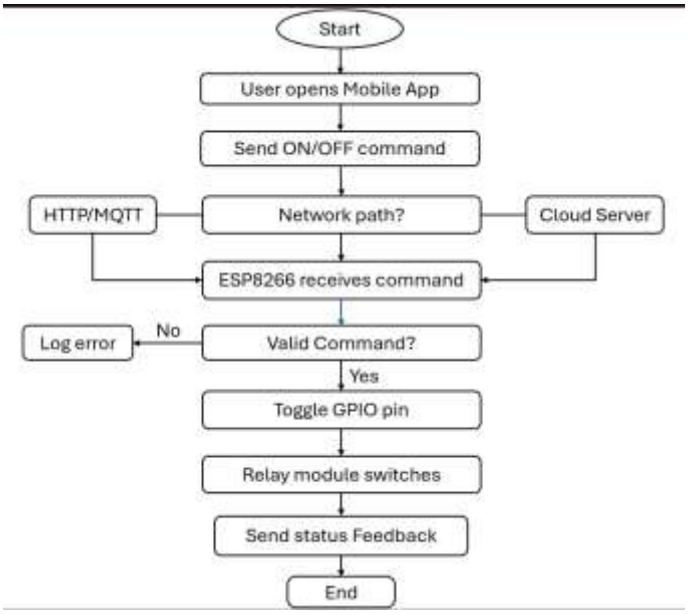

Fig. 1 — System flowchart illustrating command transmission, processing, actuation, and feedback loop

C. Hardware Design and Component Selection

The hardware architecture is centered around the NodeMCU ESP8266 microcontroller, selected due to its integrated Wi-Fi capability, low cost, compact size, and extensive software ecosystem. The ESP8266 operates at 3.3V logic levels and provides multiple GPIO pins for interfacing with external devices.

A 5V relay module with optocoupler isolation is used to interface the microcontroller with high-voltage AC loads (230V). The optocoupler ensures electrical isolation between control and power circuits, enhancing system safety. The relay is configured in a normally open (NO) mode to ensure that appliances remain off by default.

A regulated power supply provides stable voltage to both the ESP8266 and relay module. Test loads such as a 60W bulb and a 40W fan are used to emulate real-world household appliances.

D. Circuit Design and Safety Considerations

Schematic Diagram shows the circuit schematic of the proposed system. The ESP8266 GPIO pins are connected to the relay module input channels to control switching operations. GPIO4 is used as the primary control pin, while GPIO5 allows optionalexpansion for additional channels.

The relay COM and normally open (NO) terminals are connected in series with the AC supply and the load. Electrical isolation between the low-voltage control side and high-voltage load side is achieved through the relay's optocoupler mechanism.

Proper insulation, secure wiring, and separation between high-voltage and low-voltage sections are maintained to ensure safe operation. The design ensures galvanic isolation between control and load circuits, reducing the risk of electrical hazards. Although not implemented in the prototype, protective components such as fuses and surge suppressors are recommended for real-world deployment.

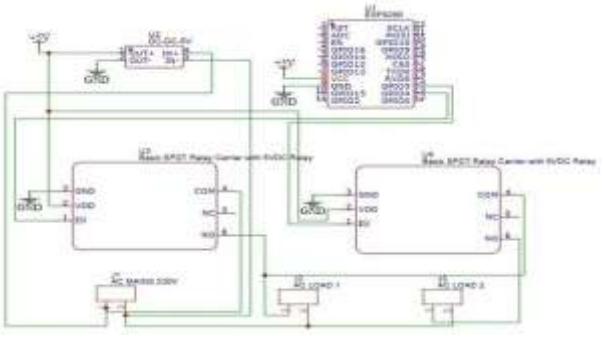

Fig. 2 — Circuit schematic of ESP8266 interfaced with relay module showing GPIO control, power supply, and isolation from AC load

E. Communication Architecture

The system employs a hybrid communication architecture combining local Wi-Fi communication with cloud-based connectivity. MQTT is used as the primary communication protocol due to its lightweight publish–subscribe mechanism, which reduces bandwidth usage and enables low-latency communication.

The ESP8266 subscribes to predefined MQTT topics hosted on a broker, allowing it to receive commands asynchronously. For remote access, the system integrates with the Blynk IoT platform, enabling control over the internet.

HTTP-based communication is also supported as a fallback mechanism, although it introduces higher latency due to its request-response nature. The dual-mode communication approach ensures system robustness and reduces dependency on continuous internet connectivity.

F. Firmware Design and Control Logic

The firmware is developed using the Arduino IDE and manages connectivity, command processing, and device control. Upon startup, the ESP8266 establishes a Wi-Fi connection and subscribes to MQTT topics corresponding to each device.

Incoming messages are parsed to determine the desired state (ON/OFF), which is then mapped to GPIO outputs. A HIGH signal activates the relay, while a LOW signal deactivates it. A watchdog timer is implemented to detect connection failures and automatically re-establish communication.

The system also sends status feedback to the mobile application

after each operation, ensuring synchronization between the physical device and the user interface.

G. Working Principle

The system operates based on user commands issued through the mobile application. When a command is generated, it is transmitted via MQTT or HTTP depending on network conditions. The ESP8266 receives and validates the command, updates the GPIO state, and triggers the relay to control the connected appliance.

Simultaneously, feedback is sent back to the application to update the device status in real time. The system supports both manual control and scheduled automation, enabling flexible operation.

H. Design Advantages

The proposed system offers several advantages:

- Low cost: Implementation cost is approximately ₹450–600
- Low latency: MQTT-based local communication ensures fast response
- Scalability: Additional devices can be integrated using multiple GPIO channels
- Reliability: Watchdog-based reconnection improves stability
- Flexibility: Supports both local and remote operation

V. Experimental Results and Analysis

The proposed system was experimentally evaluated over a 7-day period under typical residential conditions to assess its performance in terms of response time, reliability, and energy efficiency. The test setup consisted of a NodeMCU ESP8266 v3, a 5V relay module, and representative household loads including a 60W incandescent bulb and a 40W ceiling fan. The system was connected to a TP-Link Archer C6 router operating on a 2.4 GHz Wi-Fi network with an average internet speed of 20 Mbps.
Control commands were issued using the Blynk mobile application on Android 12. Measurements were conducted under four scenarios: Local MQTT, Local HTTP, Remote MQTT, and Remote HTTP. Response time was defined as the interval between issuing a command on the application and the physical actuation of the relay, observed through visual and audible confirmation.

A. Response Time Analysis

Table I presents the response time measurements recorded across 10 trials for each communication scenario.

Trial	Local MQTT	Local HTTP	Remote MQTT	Remote HTTP
1	271	318	589	724
2	265	302	601	698
3	290	335	578	741
4	281	311	623	715
5	275	324	614	752
6	298	341	598	708
7	263	308	631	689
8	285	329	607	733
9	277	315	594	718
10	279	322	585	727
Mean	278.4	321.0	602.0	720.5
Std Dev	10.5	12.3	17.2	19.4

The results indicate that Local MQTT achieved the lowest mean response time of 278.4 ms, followed by Local HTTP at 321.0 ms.Remote communication exhibited higher latency, with Remote MQTT averaging 602.0 ms and Remote HTTP reaching 720.5 ms.

The superior performance of MQTT can be attributed to its lightweight publish–subscribe architecture, which minimizes communication overhead and enables asynchronous message delivery. In contrast, HTTP incurs additional latency due to its request–response mechanism and higher protocol overhead.

The low standard deviation observed across all scenarios indicates consistent system performance with minimal variability, demonstrating stable operation under repeated trials.

B. Reliability Analysis

Table II summarizes the system reliability over a continuous 7-day testing period.

Day	Commands sent	Successful	Failed	Success rate
1	48	48	0	100%
2	52	51	1	98.1%
3	45	45	0	100%
4	50	49	1	98.0%
5	47	47	0	100%
6	53	51	2	96.2%
7	49	48	1	98.0%
Total	344	339	5	98.5%

A total of 344 commands were issued, of which 339 were successfully executed, resulting in an overall success rate of 98.5%. Daily success rates remained consistently above 96%, indicating high system stability.

The recorded failures (5 instances) were primarily due to transient Wi-Fi disconnections rather than hardware or firmware faults. In all cases, the system successfully recovered within 8–
12 seconds through the implemented watchdog-based reconnection mechanism.

These results demonstrate that the proposed system maintains reliable operation under real-world conditions with minimal failure rates.

C. Energy Efficiency Analysis

Table III presents the power consumption measurements under different operating conditions.

Condition	Power (W)	Daily (Wh)
Manual switching (baseline)	62.4	1497.6
With ESP8266 automation	53.3	1279.2
ESP8266 standby only	1.0	24.0
Relay idle draw	0.4	9.6
Net saving	9.1 W (14.6%)	218.4

The baseline power consumption under manual switching was measured at 62.4 W, whereas the automated system reduced consumption to 53.3 W. This corresponds to a net reduction of
9.1 W, equivalent to a 14.6% improvement in energy efficiency. The observed savings are primarily attributed to the elimination of unnecessary appliance usage through scheduling and remote control. The ESP8266 and relay module introduce a standby power consumption of approximately 1.4 W, which is negligible compared to the overall energy savings achieved.

These results validate the effectiveness of the proposed system in optimizing energy usage without introducing significant overhead.

D. Performance Visualization

Figure 3 presents a graphical representation of system performance across different scenarios.

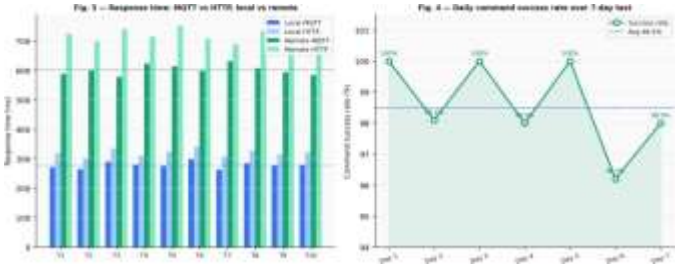

Fig. 3 — Response time comparison across communication modes (left) and daily reliability over a 7-day period (right)
The response time graph clearly illustrates the latency differences between MQTT and HTTP protocols, with MQTT consistently outperforming HTTP in both local and remote scenarios. The reliability graph shows stable system performance, with daily success rates consistently above 96% and a mean value of 98.5%.

E. Comparative Analysis with Existing Systems

Table IV compares the performance of the proposed system with existing ESP8266-based smart home automation solutions reported in the literature.

System	Protocol	Avg latency (ms)	Reliability	Cost (approx.)
Stolojescu-Crisan et al. [1]	HTTP	400–600	Not reported	Medium
Tayef et al. [2]	HTTP REST	350–500	~95%	Medium
Rawat et al. [3]	MQTT	280–400	~96%	Low
Shivdas et al. [4]	MQTT+GSM	500–900	~97%	Medium-high
Proposed system	MQTT	278 (local)	98.5%	Low (₹450–600)

The proposed system demonstrates competitive performance, achieving lower local latency (278 ms) and higher reliability (98.5%) compared to several existing implementations. Additionally, the system maintains a lower cost range (₹450–600), making it suitable for large-scale deployment.
It should be noted that performance comparisons across different studies may vary due to differences in experimental setups, network conditions, and evaluation methodologies. Nevertheless, the results indicate that the proposed system provides an effective balance between cost, performance, and reliability.

VI. Conclusion

This paper presented the design and implementation of a low-cost IoT-based smart home automation switch using the ESP8266 microcontroller. The system enables real-time monitoring and control of household appliances through Wi-Fi-based communication and a mobile application interface.

Experimental evaluation over a 7-day period demonstrated low latency (278 ms for local MQTT), high reliability (98.5% command success rate), and a 14.6% reduction in energy consumption compared to manual switching. The proposed solution offers a cost-effective (₹450–600), scalable, and efficient alternative to conventional automation systems.

However, the system is dependent on network availability and currently lacks secure communication mechanisms. Future work will focus on integrating TLS-based security, enhancing system robustness, and extending the architecture for large-scale multi-device deployments.

VII. References

[1] C. Stolojescu-Crisan, C. Crisan, and B.-P. Butunoi, “An IoT-based smart home automation system,” *Sensors*, vol. 21, no. 11, p. 3784, May 2021, doi: 10.3390/s21113784.
S. H. Tayef, M. M. Rahman, and M. A. B. Sakib, “Design and implementation of IoT based smart home automation system,” in *Proc. IEEE Int. Conf. on Recent Engineering Science and Technology (ICREST)*, Dhaka, Bangladesh, 2021, pp. 570–575, doi: 10.1109/ICREST51555.2021.9368666.

[2] J. Rawat *et al.*, “IoT-based home automation system using ESP8266,” in *Lecture Notes in Electrical Engineering*, vol. 1007, Singapore: Springer, 2023, pp. 633–643, doi: 10.1007/978-981-99-3315-0_53.

[3] D. N. Shivdas, P. Y. Sanjay, and S. S. Anwar, “IoT-based home automation system using ESP8266,” *Int. J. Emerging Directions in Research*, vol. 1, Feb. 2026, Art. no. IJEDR2601337.

[4] A. Adhikary *et al.*, “Design and implementation of an IoT-based smart home automation system in real-world scenario,” *Int. J. Advanced Computer Science and Applications*, vol. 15, no. 3, 2024.

[5] M. A. Al-Garadi *et al*., "A survey of machine and deep learning methods for IoT security," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1646–1685, 2023, doi: 10.1109/COMST.2022.3214364.

[6] D. Minoli, K. Sohraby, and B. Occhiogrosso, "IoT considerations for smart buildings: Energy optimization and next-generation building management systems," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 19877–19905, Oct. 2022, doi: 10.1109/JIOT.2022.3158817.

[7] R. Ammar, B. Akhter, and T. Whangbo, "NodeMCU ESP8266-based smart energy monitoring with cloud integration," *Electronics*, vol. 12, no. 5, p. 1208, Mar. 2023, doi: 10.3390/electronics12051208.

[8] H. Li, K. Ota, and M. Dong, "Learning IoT in edge: Deep learning for IoT with edge computing," *IEEE Network*, vol. 38, no. 1, pp. 96–101, Jan. 2024, doi: 10.1109/MNET.2023.3321567.

[9] F. Al-Turjman, M. H. Nawaz, and U. D. Ulusar, "Intelligence in the IoT era: A systematic review," *Information Fusion*, vol. 100, p. 101904, Dec. 2023, doi: 10.1016/j.inffus.2023.101904.

[10] Y. Yang *et al*., "A survey on security and privacy issues in IoT," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 3899–3911, Mar. 2023, doi: 10.1109/JIOT.2022.3213902.

[11] P. Arora and S. Bhatt, "Real-time home automation using ESP8266 and MQTT protocol," *Int. J. Innovative Technology and Exploring Engineering*, vol. 11, no. 4, pp. 78–84, Feb. 2022, doi: 10.35940/ijitee.C9813.0211422.

[12] W. Shi *et al*., "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 3001–3015, Jan. 2024, doi: 10.1109/JIOT.2023.3311426.

[13] N. Kshetri, "Cybersecurity and privacy issues in cloud-connected IoT devices," *Computer*, vol. 56, no. 1, pp. 10–19, Jan. 2023, doi: 10.1109/MC.2022.3216545.

[14] K. Suresh Kumar *et al*., "Smart plug design using ESP8266 NodeMCU for home energy monitoring," *Journal of Physics: Conference Series*, vol. 2318, no. 1, p. 012003, 2022, doi: 10.1088/1742-6596/2318/1/012003.

[15] S. S. Gill *et al*., "Transformative effects of IoT, Blockchain and AI on cloud computing," *Internet of Things*, vol. 25, p. 100893, Mar. 2024, doi: 10.1016/j.iot.2023.100893.

[16] M. S. Munoz-Abad *et al*., "Design of an IoT gateway for distributed smart home systems," *Sensors*, vol. 23, no. 14, p. 6402, Jul. 2023, doi: 10.3390/s23146402.

[17] A. Qayyum *et al*., "Securing connected and autonomous vehicles: Challenges posed by adversarial machine learning," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 460–501, 2024, doi: 10.1109/COMST.2023.3329328.

[18] R. Verma, P. Sharma, and A. K. Singh, "Energy-efficient smart home automation using ESP32 and MQTT with TLS," *Measurement: Sensors*, vol. 32, p. 101022, Apr. 2025, doi: 10.1016/j.measen.2024.101022.

[19] T. N. Nguyen, B. M. Le, and V. H. Nguyen, "Lightweight authentication protocol for IoT smart home devices using ESP8266," *Applied Sciences*, vol. 15, no. 3, p. 1544, Feb. 2025, doi: 10.3390/app15031