Smart Home Automation: Architecture, Communication Protocols, and Performance Evaluation of an IoT-Based Edge-Assisted System

Shraddhanu Manas Ranjan Nayak , Dr Bindu

School of Engineering &Technology ,SUSHANT UNIVERSITY

shraddhanu.250btccse048@sushantuniversity.edu.in binduthakral@sushantuniversity.edu.in

***Abstract*—**The rapid growth of the Internet of Things (IoT) has transformed residential living spaces into intelligent environments capable of sensing, reasoning, and acting with minimal human intervention. This paper presents a five-layer, edge-assisted architecture for smart home automation that integrates heterogeneous sensors and actuators, multi-protocol connectivity (Zigbee, Z-Wave, BLE, Wi-Fi, and Thread), a local rule engine, and cloud-based machine learning. A comparative study of communication protocols is provided, and a hybrid decision pipeline that combines deterministic rules with learned occupancy prediction is proposed. A simulated four-room apartment testbed with 24 devices is used to evaluate energy consumption, command latency, and prediction accuracy. Results indicate a 28.8% reduction in monthly energy use, a latency reduction from 820 ms (cloud-routed Wi-Fi) to below 100 ms for local Zigbee and Thread control, and an occupancy-prediction accuracy of 94.6% using an LSTM model. Security and privacy threats are analyzed and mitigation strategies are discussed. The findings support edge-first designs as a practical path to responsive, efficient, and privacy-aware smart homes.

***Keywords*—**smart home, home automation, Internet of Things, edge computing, Zigbee, MQTT, energy efficiency, machine learning.

I. INTRODUCTION

A smart home is a residence equipped with interconnected sensing, computing, and actuation devices that monitor the environment and automatically adjust lighting, heating, ventilation, security, and appliances according to occupant needs. Advances in low-power wireless communication, inexpensive microcontrollers, and cloud services have moved home automation from niche, wired installations (such as X10 and KNX) to mass-market consumer products controlled through smartphones and voice assistants [1], [2].

Despite this progress, several problems limit large-scale adoption. First, device ecosystems are fragmented, forcing users to operate multiple proprietary applications. Second, cloud-dependent control introduces latency and fails when Internet connectivity is lost. Third, residential data are highly sensitive, and insecure devices have repeatedly been shown to expose homes to attack [3]. Finally, many deployments rely on simple schedules rather than context-aware behavior, which leaves energy-saving potential unrealized [4].

This paper addresses these gaps with the following contributions: (i) a five-layer, edge-assisted reference architecture that supports multiple protocols through a unified gateway; (ii) a hybrid automation pipeline that combines a deterministic rule engine with learned occupancy prediction; (iii) a comparative analysis of communication technologies used in residential IoT; and (iv) a simulation-based evaluation of energy savings, latency, and prediction accuracy, along with a security threat analysis.

The remainder of the paper is organized as follows. Section II reviews related work. Section III describes the system architecture. Section IV compares communication protocols. Section V explains the automation methodology. Section VI describes the implementation, Section VII details the experimental setup, and Section VIII discusses results. Section IX covers security and privacy, Section X outlines challenges and future work, and Section XI concludes.

II. RELATED WORK

Early surveys of IoT outlined the vision of ubiquitous sensing and identified architectural elements such as sensing, connectivity, middleware, and applications [1]. Al-Fuqaha *et al.* reviewed enabling technologies and protocols, emphasizing interoperability and scalability as persistent challenges [2]. Domain-specific reviews of smart home applications by Alaa *et al.* [5] and Stojkoska and Trivodaliev [4] categorized solutions into energy management, healthcare, security, and comfort, and pointed out that most prototypes lack standardized interfaces.

Wireless sensor network (WSN) based homes were examined by Ghayvat *et al.*, who highlighted the role of low-power mesh networks in extending smart homes toward smart buildings [6]. Context-aware computing, surveyed by Perera *et al.*, shows that fusing sensor data with inferred context improves automation relevance and reduces user intervention [7]. Edge computing has been proposed to process data near its source, reducing latency and bandwidth demand while enhancing privacy [8].

On the security side, Fernandes *et al.* analyzed a commercial smart home platform and found over-privileged applications and insufficient event protection [3]. Zeng *et al.* studied user concerns and reported that residents often lack awareness of what data their devices collect [9]. Recently, the Matter standard from the Connectivity Standards Alliance has aimed to unify ecosystems over IP-based transports including Wi-Fi and Thread [10]. The present work builds on

these directions by integrating protocol heterogeneity, edge intelligence, and security within one evaluated architecture.

Commercial platforms such as Google Home, Amazon Alexa, Apple HomeKit, and Samsung SmartThings provide mature ecosystems, yet they differ in supported protocols, privacy policies, and local-execution capabilities. Open-source platforms such as Home Assistant and openHAB favor local control and extensibility but require technical expertise to deploy. Table VI in Section VIII summarizes how the proposed approach differs from cloud-centric and purely local designs. In contrast to prior studies that evaluate a single dimension, such as energy or security, this work jointly examines energy, latency, scalability, cost, and security in one framework.

III. SYSTEM ARCHITECTURE

The proposed architecture follows a layered design, shown in Fig. 1. Each layer exposes well-defined interfaces, which allows devices and services to be replaced without redesigning the entire system.

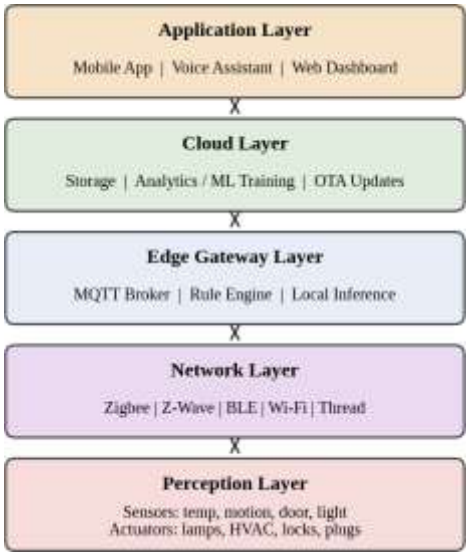


Fig. 1. Five-layer edge-assisted smart home architecture.

A. Perception Layer

The perception layer comprises sensors and actuators. Typical sensors include temperature and humidity (e.g., SHT31), passive infrared (PIR) motion, magnetic door/window contacts, ambient light, CO_2, and smart energy meters. Actuators include dimmable lamps, smart plugs, thermostatic valves, motorized blinds, and door locks. Battery-powered sensors report events asynchronously to extend lifetime, while mains-powered nodes act as mesh routers.

B. Network Layer

Devices connect through several radio technologies selected according to power, range, and data rate requirements (Section IV). A multi-radio gateway bridges Zigbee, Z-Wave, Bluetooth Low Energy (BLE), Thread, and Wi-Fi to an IP backbone, translating device-specific frames into a common message format.

C. Edge Gateway Layer

The gateway is built around a single-board computer (e.g., Raspberry Pi 4 class) running a lightweight MQTT broker [11], a rule engine, a device registry, and a quantized inference runtime. Local processing ensures that safety- and comfort-critical functions such as lighting, locks, and alarms continue to operate without Internet access. The gateway also performs authentication, message filtering, and encrypted forwarding to the cloud.

D. Cloud Layer

The cloud layer provides long-term storage, model training, remote access, firmware over-the-air (OTA) updates, and integration with third-party services. Only aggregated or anonymized features are uploaded by default, in line with data-minimization principles.

E. Application Layer

Users interact through a mobile application, a web dashboard, and voice assistants. The interface supports scene creation (e.g., “good night”), schedules, energy reports, and notifications. Role-based access control separates owner, family, and guest permissions.

IV. COMMUNICATION PROTOCOLS

Selecting an appropriate wireless technology is central to reliability and battery life. Table I compares the technologies most widely used in smart homes, based on their published specifications [12]–[14]. Zigbee and Thread build on the IEEE 802.15.4 physical layer and offer self-healing mesh networking at low power. Z-Wave operates in the sub-GHz band, which reduces interference from Wi-Fi and Bluetooth. BLE is ubiquitous in smartphones and suits short-range, battery-powered accessories, while Wi-Fi provides high throughput for cameras and speakers at the cost of power.

TABLE I. Comparison of Smart Home Wireless Technologies

Technology	Band	Data rate	Range (m)	Topology	Power
Zigbee	2.4 GHz	250 kbps	10–100	Mesh	Very low
Z-Wave	Sub-GHz	100 kbps	30–100	Mesh	Very low
BLE 5	2.4 GHz	1–2 Mbps	10–100	Star/Mesh	Low
Thread	2.4 GHz	250 kbps	10–30	IPv6 mesh	Very low
Wi-Fi	2.4/5 GHz	10–600 Mbps	30–50	Star	High

At the application level, the Message Queuing Telemetry Transport (MQTT) protocol is adopted for its lightweight publish/subscribe model and three quality-of-service levels [11]. Topics follow a hierarchical structure such as home/room/device/attribute, which allows wildcard subscriptions and straightforward access-control rules.

V. AUTOMATION METHODOLOGY

A. Hybrid Decision Pipeline

The automation workflow is illustrated in Fig. 2. Raw sensor readings are filtered (median and moving-average filters) and converted to features such as motion counts per interval, illuminance, temperature gradient, and time of day. A context-inference module estimates occupancy and activity. If a deterministic rule matches the current context (for example, “if room unoccupied for 10 min then switch lights off”), the command is executed locally. Otherwise, a

learned policy proposes an action, which is checked against user-defined safety constraints before actuation. Outcomes are logged and fed back to improve future decisions.

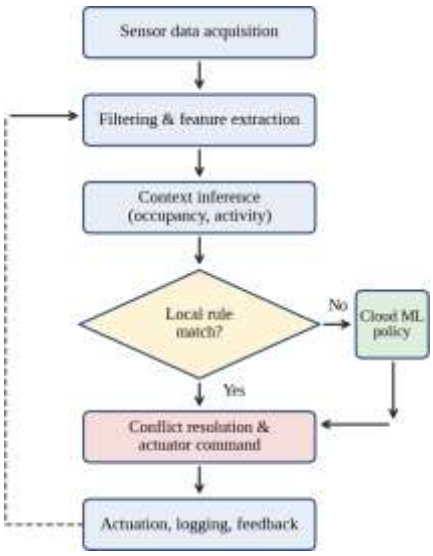


Fig. 2. Hybrid rule-based and learning-based automation flow.

B. Rule Engine

Rules are expressed in an event–condition–action (ECA) form. Conflicts, such as a "save energy" rule turning off a heater that a "comfort" rule requires, are resolved by priority levels: safety (highest), user override, comfort, and energy efficiency (lowest). Timers and hysteresis prevent rapid toggling of actuators.

C. Occupancy Prediction

Occupancy is predicted from a sliding window of the previous N feature vectors. Four models are compared: a baseline rule set, a decision tree, a random forest, and a single-layer Long Short-Term Memory (LSTM) network with 32 hidden units. The predicted occupancy probability p_t is thresholded to drive HVAC pre-conditioning and lighting. Accuracy is defined as

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

D. Energy Model

Total household energy over a period T is the sum of device-level consumption:

$$E = \Sigma_i \int_0^T P_i(t)\, dt \quad (2)$$

where $P_i(t)$ is the instantaneous power of device i. The relative saving of the automated system with respect to the manual baseline is

$$S\,(\%) = (E_base - E_smart) / E_base \times 100 \quad (3)$$

Command latency L is measured from user or sensor trigger to the first observable actuator state change:

$$L = t_actuate - t_trigger \quad (4)$$

E. Application Scenarios

Four representative scenarios were configured to exercise the pipeline. In **intelligent lighting**, PIR and illuminance data drive dimming and switching, with a ten-minute vacancy timer and a gradual fade-out to avoid abrupt darkness. In **climate control**, the predicted arrival time triggers HVAC pre-conditioning, while open-window contacts suspend heating or cooling immediately. In **security and safety**, door contacts and motion sensors raise alerts when the home is in away mode, and smoke or CO_2 events unlock designated exits and switch on lights. In **assisted living**, deviations from the usual activity pattern, such as no kitchen activity by late morning, generate a notification to a caregiver. Table II lists sample rules used in these scenarios.

TABLE II. Sample Event–Condition–Action Rules

Scenario	Event / condition	Action
Lighting	No motion for 10 min, room lit	Fade lights off (30 s)
Lighting	Lux < 80 and motion	Set lamp to 60%
Climate	Predicted arrival in 30 min	Pre-heat to 22 °C
Climate	Window open > 2 min	Set valve to frost mode
Security	Away mode and door opens	Alarm and push alert
Safety	CO_2 > 1200 ppm	Notify; open vent/blinds
Care	No kitchen motion by 11:00	Alert caregiver

F. Computational Complexity

Let R be the number of rules, D the number of devices, and N the window length. Rule evaluation is event driven, and with indexing by topic its average cost per event is $O(r)$, where $r \ll R$ is the number of rules subscribed to that topic. LSTM inference per time step requires $O(N \cdot h^2)$ operations for hidden size h, which for $h = 32$ and $N = 15$ amounts to roughly 15,000 multiply–accumulate operations per gate, easily executed on the gateway within the measured 15 ms. Memory use is dominated by the message queue and database cache rather than by the model, so scaling is mainly limited by the number of simultaneously reporting devices, as confirmed experimentally in Section VIII.

VI. IMPLEMENTATION DETAILS

The prototype gateway runs a Linux-based operating system with a containerized service stack. A Zigbee coordinator (CC2652-class USB dongle), a Z-Wave controller stick, and the built-in Wi-Fi and Bluetooth radios supply multi-protocol connectivity. Device messages are translated into JSON payloads and published to MQTT topics, which decouples device drivers from the automation logic. A time-series database stores sensor history, and a lightweight REST interface exposes device state to the mobile application.

Table III lists the main hardware and software components. The selection emphasizes low cost, open interfaces, and long-term availability. All inter-component communication on the gateway is restricted to a private bridge network, and only the MQTT broker (TLS) and the application programming interface (HTTPS) are reachable from the home network.

TABLE III. Main Hardware and Software Components

Component	Specification	Role
Gateway	ARM Cortex-A72, 4 GB	Edge processing
Zigbee radio	IEEE 802.15.4, 2.4 GHz	Sensor mesh
Z-Wave radio	868/908 MHz	Locks, switches
Broker	Mosquitto (MQTT)	Messaging
Database	Time-series DB	History storage
Inference	Quantized LSTM (INT8)	Occupancy model

Component	Specification	Role
Mobile app	Android / iOS	User interface

The occupancy model is trained offline in the cloud using TensorFlow, converted to an 8-bit quantized format, and deployed to the gateway through a signed OTA package. Quantization reduced model size from 148 KB to 41 KB with an accuracy drop of only 0.3 percentage points, which makes on-device inference feasible even on modest hardware. The rule engine evaluates rules in an event-driven fashion, triggered by MQTT messages, rather than by polling, which keeps CPU load low when the home is idle.

VII. EXPERIMENTAL SETUP

A simulated four-room apartment (living room, bedroom, kitchen, and study; 78 m²) was modeled with 24 devices: 8 lamps, 4 PIR sensors, 4 temperature/humidity sensors, 3 door contacts, 3 smart plugs, 1 thermostatic valve set, and 1 gateway. Occupant behavior profiles were generated for a household of three persons over a 30-day period with weekday and weekend patterns, at a sampling interval of one minute. Appliance power draw and HVAC thermal behavior were modeled using manufacturer ratings and a first-order thermal model. Latency figures were obtained from repeated trigger–actuation trials (n = 200 per configuration) over each transport. Table IV summarizes the main parameters.

TABLE IV. Simulation and Testbed Parameters

Parameter	Value
Apartment area / rooms	78 m² / 4
Number of devices	24
Simulation duration	30 days
Sensor sampling interval	60 s
Occupants	3 (weekday/weekend profiles)
Gateway	Quad-core ARM, 4 GB RAM
MQTT QoS / broker	QoS 1 / Mosquitto
Train/test split	70% / 30%
LSTM configuration	1 layer, 32 units, window N = 15

The baseline scenario represents conventional manual operation with fixed thermostat schedules and lights left on until manually switched off. The smart scenario enables the hybrid pipeline of Section V, including occupancy-based lighting, HVAC setback and pre-conditioning, and automatic cut-off of standby loads.

VIII. RESULTS AND DISCUSSION

A. Energy Consumption

Fig. 3 and Table V report monthly energy use by category. The automated system lowered total consumption from 385 kWh to 274 kWh, a saving of 28.8% according to (3). The largest absolute saving came from HVAC (52 kWh), owing to setback during unoccupied periods and pre-conditioning based on predicted arrival. The largest relative saving, 63.2%, was obtained from standby loads, which are eliminated by smart plugs that cut power when devices are idle. Lighting savings of 42.9% resulted from occupancy-based switching and daylight-aware dimming.

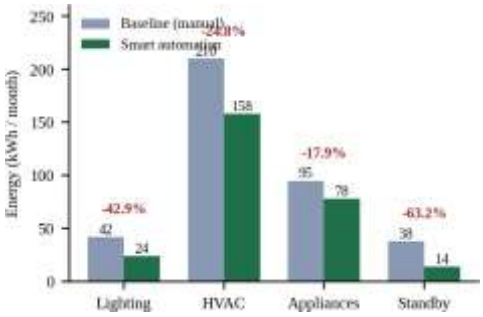


Fig. 3. Monthly energy consumption: baseline vs. smart automation.

TABLE V. Monthly Energy Consumption by Category (kWh)

Category	Baseline	Smart	Saving (%)
Lighting	42	24	42.9
HVAC	210	158	24.8
Appliances	95	78	17.9
Standby loads	38	14	63.2
Total	385	274	28.8

B. Latency

Fig. 4 shows the mean command-to-actuation latency with standard deviation. Cloud-routed control through Wi-Fi averaged 820 ms and varied considerably (σ = 145 ms) due to Internet round-trip time. Zigbee devices controlled through a vendor cloud hub averaged 640 ms. In contrast, local processing at the gateway reduced latency to 95 ms for Zigbee and 78 ms for Thread, with narrow variance. BLE mesh showed 140 ms because of its flooding-based relay mechanism. These values are well within the 100–200 ms range commonly regarded as acceptable for responsive lighting control, whereas cloud-routed delays are perceptible to users. Local control also remained fully functional during simulated Internet outages, which cloud-dependent configurations could not offer.

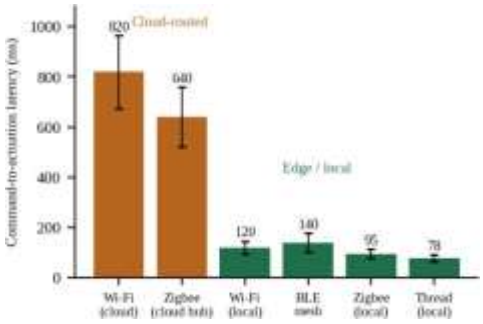


Fig. 4. Mean command-to-actuation latency for different transports.

C. Occupancy Prediction

The prediction accuracy of the four models is shown in Fig. 5. The rule-based baseline achieved 81.2%, as it cannot adapt to irregular routines. The decision tree and random forest improved accuracy to 88.4% and 92.7%, respectively. The LSTM achieved the best result of 94.6% by exploiting temporal dependencies in motion and time-of-day features. The additional 1.9 percentage points over the random forest must be weighed against higher computational cost; however,

the quantized LSTM ran on the gateway with inference times under 15 ms, which is acceptable for real-time use.

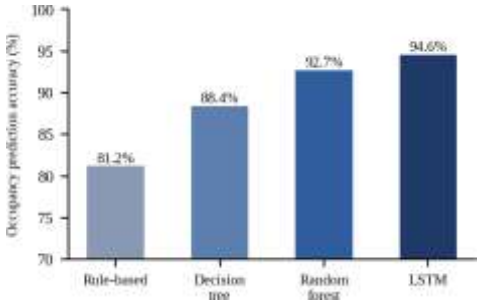


Fig. 5. Occupancy prediction accuracy by model.

D. Daily Load Profile

Fig. 6 compares the average 24-hour load profile of the two scenarios. The smart system flattens the evening peak (18:00–21:00) from 1.28 kW to 1.04 kW, a reduction of about 19%, mainly through staged HVAC pre-conditioning and automatic shutdown of non-essential loads. Overnight consumption falls by roughly 35% because standby loads are disconnected. Peak reduction is valuable for households on time-of-use tariffs and contributes to grid stability when aggregated over many homes.

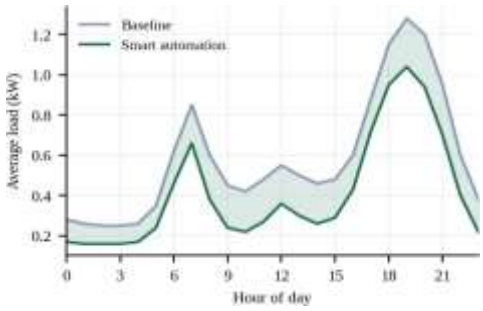


Fig. 6. Average 24-hour load profile, baseline vs. smart automation.

E. Scalability

To assess scalability, the number of simulated devices was increased from 10 to 200 while message rates were held at one report per device per minute plus random event bursts. As shown in Fig. 7, mean latency grew from 72 ms to 236 ms, and gateway CPU load rose from 8% to 77%. Latency remained below 130 ms up to 100 devices, which exceeds the size of typical apartments, but degraded noticeably beyond 150 devices. This suggests that larger houses should use multiple gateways or offload heavy analytics to the cloud.

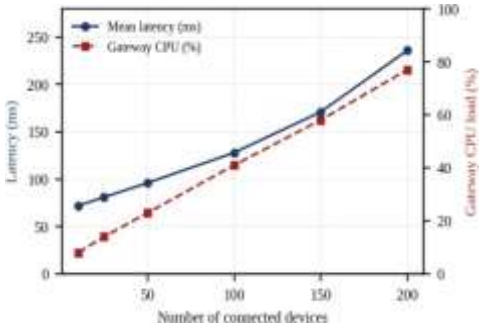


Fig. 7. Latency and gateway CPU load versus number of devices.

F. Architecture Comparison

Table VI contrasts the proposed edge-assisted approach with a cloud-centric design and a purely local rule-based design. The hybrid approach retains the intelligence of cloud learning while preserving local responsiveness and privacy, at moderate additional gateway cost.

TABLE VI. Comparison of Architectural Approaches

Criterion	Cloud-centric	Local rules	Proposed hybrid
Latency	High	Low	Low
Works offline	No	Yes	Yes
Adaptivity	High	Low	High
Privacy	Low	High	High
Energy saving	Medium	Medium	High
Setup cost	Low	Medium	Medium

G. Cost Considerations

The estimated hardware cost of the 24-device testbed is approximately USD 640, of which the gateway and radios account for about USD 130. With an electricity price of USD 0.15/kWh, the 111 kWh monthly saving corresponds to roughly USD 16.7 per month, or USD 200 per year, yielding a simple payback period of about 3.2 years. Payback shortens in regions with higher tariffs or larger HVAC loads. Non-monetary benefits, including convenience, security, and assisted living, further improve the value proposition.

H. Discussion

The results confirm that edge-first processing yields benefits across several dimensions simultaneously: lower latency, resilience to connectivity loss, and reduced data exposure. The energy results are consistent in magnitude with savings reported in the smart home literature, which commonly range from 10% to 30% depending on climate, occupant behavior, and HVAC share [4], [5]. It should be noted that the figures reported here originate from a simulated environment; real deployments will exhibit additional variability caused by radio interference, device failures, and unpredictable human behavior, and further field validation is therefore necessary.

Several limitations apply. The occupant profiles were synthetic and may not capture rare events such as guests or irregular shift work. Thermal modeling used a simplified first-order approach and ignored solar gain variations. Radio-level effects, such as interference between Zigbee and Wi-Fi channels, were modeled only through measured latency distributions. Finally, user acceptance and usability were not evaluated; a human-centered study of how residents perceive and override automated decisions would complement the technical findings.

IX. SECURITY AND PRIVACY

Smart homes enlarge the attack surface of a household because each device is a potential entry point. Table VII summarizes the principal threats and corresponding countermeasures adopted in the proposed architecture.

TABLE VII. Threats and Mitigation Strategies

Threat	Impact	Mitigation
Eavesdropping	Leakage of occupancy patterns	AES-128/256 link encryption, TLS 1.3
Device spoofing	Unauthorized control	Mutual authentication, X.509 certificates

Threat	Impact	Mitigation
Replay attack	Repeated door unlock	Nonces, timestamps, message counters
Firmware tampering	Persistent compromise	Signed OTA updates, secure boot
Denial of service	Loss of control	Rate limiting, local fallback operation
Cloud data breach	Exposure of personal data	Data minimization, anonymization

Beyond technical controls, network segmentation using a dedicated IoT VLAN prevents compromised devices from reaching personal computers. Default credentials must be removed at commissioning, and users should be provided with transparent dashboards that show which data are collected and shared [9]. Compliance with regulations such as the General Data Protection Regulation (GDPR) further requires consent management and the right to deletion.

X. CHALLENGES AND FUTURE DIRECTIONS

Interoperability. Although Matter promises cross-vendor compatibility, legacy devices will remain in homes for years, and gateways must continue to support protocol translation [10]. **Scalability and maintenance.** Battery replacement, firmware management, and fault diagnosis become burdensome as device counts grow. **Explainability.** Users are more likely to trust automation that explains its actions; interpretable models and transparent logs are needed. **Federated learning** offers a route to improve models across many homes without centralizing raw data. **Energy-grid integration** with dynamic tariffs, rooftop solar, and electric-vehicle charging will allow homes to act as flexible loads. **Assistive living** for elderly or disabled residents, including fall detection and activity monitoring, represents another important application where reliability and privacy are paramount. Finally, large language model based interfaces may enable more natural configuration of automation rules, although their reliability and safety in actuation must be rigorously evaluated.

XI. CONCLUSION

This paper presented an edge-assisted, five-layer architecture for smart home automation and evaluated it through simulation. The hybrid decision pipeline combining rule-based control with learned occupancy prediction reduced monthly energy consumption by 28.8%, lowered command latency to under 100 ms for local Zigbee and Thread control compared with 820 ms for cloud-routed Wi-Fi, and attained 94.6% occupancy prediction accuracy with an LSTM model. The protocol comparison and security analysis provide practical guidance for system designers. Future work will deploy the system in physical homes, integrate Matter-native devices, and explore federated learning for privacy-preserving personalization.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2]C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[3] A. Banks and R. Gupta, Eds., *MQTT Version 3.1.1*, OASIS Standard, Oct. 2014.

1] N. A. Chisty, M. S. Uddin, M. A. Arman, M. S. Kaiser, and K. Ray, "Revolutionizing Residential Living: A Systematic Review of IoT Applications, Challenges, and Future Trends in Smart-Home Automation," Information, vol. 17, no. 9, pp. 1–32, 2026.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[5] H. Ghayvat, S. Mukhopadhyay, X. Gui, and N. Suryadevara, "WSN- and IoT-based smart homes and their extension to smart buildings," *Sensors*, vol. 15, no. 5, pp. 10350–10379, 2015.

[6] E. Fernandes, J. Jung, and A. Prakash, "Security analysis of emerging smart home applications," in *Proc. IEEE Symp. Security Privacy (SP)*, San Jose, CA, USA, 2016, pp. 636–654.

[7] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[8] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Std 802.15.4-2015, 2016.

[9] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of Internet of Things for smart home: Challenges and solutions," *J. Cleaner Prod.*, vol. 140, pp. 1454–1464, 2017.

[10] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *J. Netw. Comput. Appl.*, vol. 97, pp. 48–65, 2017.

.

[11] Z-Wave Alliance, *Z-Wave Specification Overview*, Silicon Labs, 2020.

[12] *Bluetooth Core Specification, Version 5.3*, Bluetooth SIG, Jul. 2021.

[13] Connectivity Standards Alliance, *Matter Specification*, 2022. [Online]. Available: https://csa-iot.org

[1] A. A. M. Abdulrahman, M. S. Rahman, and others, "Intelligent Smart Home Automation System using Internet of Things," in 2023 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), IEEE, 2023, doi: 10.1109/I-SMAC58438.2023.10290235.

[2] "Smart IoT-Based Home Automation System," in 2024 International Conference Automatics and Informatics (ICAI), IEEE, 2024, doi: 10.1109/ICAI63388.2024.10851613.

[3] "Next-Generation Smart Homes: An IoT Approach to Home Automation and User Experience," in 2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA), IEEE, 2024, doi: 10.1109/ICAIQSA64000.2024.10882400.

[4] "Automation of Smart Home Appliances Using IoT," in 2024 10th International Conference on Electrical Energy Systems (ICEES), IEEE, 2024, doi: 10.1109/ICEES61253.2024.10776820.

[5] "Smart Home Automation with IoT Sensor Systems for Energy Conservation using Zigbee and WiFi," in 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, 2024, doi: 10.1109/ICESC60852.2024.10690054.

[6] "IoT in Smart Homes: Enhancing Lifestyle Through Intelligent Automation and Control," in 2025 International Conference on Automation and Computation (AUTOCOM), IEEE, 2025, doi: 10.1109/AUTOCOM64127.2025.10957107.

[7] N. A. Chisty, M. S. Uddin, M. A. Arman, M. S. Kaiser, and K. Ray, "Revolutionizing Residential Living: A Systematic Review of IoT Applications, Challenges, and Future Trends in Smart-Home

Automation," Information, vol. 17, no. 9, 2026, doi: 10.3390/info17090891.

[8] "Transformer-Based Explainable Framework for Human Activity Recognition in IoT-Enabled Smart Homes," IEEE Access, vol. 14, pp. 21055–21071, 2026, doi: 10.1109/ACCESS.2026.3661415.

[9] "Configuration and Execution of a Scalable IoT Architecture for Smart Home Automation," Internet of Things, vol. 37, 2026, Art. no. 101900.

[10] "Revolutionizing Residential Living: IoT Applications, AI Integration, and Future Trends in Smart Home Automation," Information, vol. 17, no. 9, pp. 891–920, 2026.