IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156113

A Secure, Scalable, and Intelligent IoT-Based Smart Home Automation Framework Using Edge Computing for Real-Time Monitoring and Energy Optimization

Akshith Kumar M[1], Chandan R[2], Darshan G B[3]

Students, Department of Master of Computer Applications, SJB Institute of Technology, Bangalore, India[1,2,3]

Abstract: The widespread adoption of IoT technology has quietly reshaped how we think about the homes we live in. What was once a collection of isolated appliances is now becoming a network of communicating, sensing, and responding systems. Yet for all this progress, most smart home architectures still lean heavily on cloud servers to do the heavy lifting — and that creates real problems. When every sensor reading has to travel to a distant data centre before a decision is made, the delays add up, privacy is harder to guarantee, and the system becomes brittle whenever the internet connection is unreliable.

This paper presents SESI-EH — a Secure, Edge-based, Scalable, and Intelligent smart home framework built around edge computing. Rather than routing everything to the cloud, the system moves intelligence to the edge: a Raspberry Pi 4 gateway handles real-time anomaly detection, energy scheduling, and actuation locally, while ESP32 microcontrollers form the sensing and control layer. Lightweight machine learning models — Isolation Forest, an LSTM Autoencoder, and an SVM classifier — run directly on the edge device, achieving an ensemble intrusion detection accuracy of 96.3% with a total end-to-end response time under 46 ms. Energy experiments over 30 days showed a 52.8% reduction in daily consumption and an 81.2% cut in cloud-bound network traffic. These results suggest that bringing intelligence closer to home is not just technically feasible — it is the more practical path forward.

Keywords: Internet of Things (IoT), Smart Home Automation, Edge Computing, Real-Time Monitoring, Energy Optimization, Machine Learning, Anomaly Detection, Home Security, Raspberry Pi, ESP32, Intelligent Automation.

1. INTRODUCTION

Walk into almost any modern home and you will find evidence of the so-called smart home revolution — voice assistants, app-controlled lighting, connected thermostats, camera doorbells. The Internet of Things has made all of this possible, and the pace of adoption has been striking. Yet beneath the polished consumer experience lies an architectural choice that most users never think about: nearly all the intelligence in these systems lives far away, on servers owned by technology companies, accessed over the internet.

That dependence on the cloud is not a trivial design detail. When you ask your smart thermostat to adjust the temperature, your request typically leaves your home, travels to a data centre, gets processed, and then a command is sent back. On a good day with a fast connection, this takes a fraction of a second. On a bad day — or in a security emergency — it can take long enough to matter. Researchers have consistently flagged high latency, bandwidth congestion, and privacy exposure as the three major weaknesses of cloud-centric smart home designs. And yet, the field has been slow to move away from this model, in part because running sophisticated AI models on cheap embedded hardware has historically been very hard to do.

That is changing. Modern microcontrollers like the ESP32 and single-board computers like the Raspberry Pi 4 have enough processing power to run lightweight machine learning models in real time. This creates an opportunity: instead of sending raw sensor streams to the cloud, we can process them at the edge, make local decisions, and only push summarised, non-sensitive data to the cloud for long-term storage and model retraining. The result is a system that is faster, more private, and more resilient to internet outages.

This paper describes exactly such a system. We call it SESI-EH — Secure, Edge-based, Scalable, and Intelligent Edge-Home. It is a three-tier framework built on ESP32 sensor nodes, a Raspberry Pi 4 edge gateway, and a lightweight cloud

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156113

backend. The key contributions are: (a) a unified architecture that addresses security, energy management, and real-time monitoring together rather than in isolation; (b) a parallel AI inference pipeline running Isolation Forest, an LSTM Autoencoder, and an SVM classifier concurrently on the edge device; and (c) experimental validation across 30 days of real-world testing demonstrating substantial improvements over baseline cloud-dependent systems.

2. LITERATURE REVIEW

The smart home literature has grown substantially over the past decade, but a close reading reveals a recurring pattern: papers tend to solve one problem well while leaving others largely unaddressed. The table below summarises the five most relevant prior works and their limitations — limitations that directly motivated the design choices behind SESI-EH.

SL	Title	Authors & Year	Methodology	Findings	Limitations
1	IoT-Based Intelligent Smart Home Control System	Taiwo (2021)	ESP8266/ESP32-CAM, Android app, SVM-based intrusion detection	Integrated appliance control, environmental monitoring, motion detection, and intrusion classification	Heavy cloud dependence raises latency and privacy concerns
2	Enhanced Smart Home Security via Deep Learning	Ezugwu (2022)	CNN + ESP32-CAM + IoT sensors, Android monitoring	~99.8% accuracy distinguishing occupants from intruders via motion-pattern analysis	Struggles with simultaneous activities or complex behavioral scenarios
3	Real-Time Smart Home Management with Energy Saving	Elkholy (2022)	Raspberry Pi, HEMS, solar resources, Alexa voice control	23.13% cost reduction and 18.161 kWh energy savings through intelligent automation	Minimal focus on security and anomaly detection
4	Edge AI for Real-Time Anomaly Detection	Reis (2025)	Raspberry Pi, Isolation Forest, LSTM Autoencoder, Federated Learning	93.6% anomaly detection accuracy with sub-50 ms inference at the edge	Narrowly focused on anomaly detection, not full home automation
5	Edge-Based Real-Time Energy Monitoring	Huang (2025)	Edge-computing framework for real-time energy monitoring and distributed management	Demonstrated effective energy monitoring via edge-enabled intelligent strategies	Does not address security or broader smart home automation

Reading across these five studies, three things stand out. First, every paper that relies on cloud processing reports latency numbers that would be unacceptable in a genuine security emergency — 95 ms of cloud inference time is not fast enough when an intruder is at the door. Second, the papers that do achieve low latency (Reis, 2025) do so by narrowing their scope to a single task, abandoning the holistic automation vision. Third, no existing work integrates energy optimisation, intrusion detection, and anomaly monitoring into a single edge-deployed framework. SESI-EH is designed to close all three of these gaps simultaneously.

3. PROBLEM STATEMENT

To understand why a new framework is needed, it helps to think concretely about what breaks when smart homes rely on the cloud. Consider a household where every ESP32 sensor node sends its readings to AWS or Azure for processing. Under normal conditions — stable broadband, lightly loaded cloud servers, no security events — this works reasonably

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156113

well. But consider what happens at 2:30 AM when an intruder approaches the front door, the broadband router is under heavy load from a neighbourhood software update, and the cloud server is handling thousands of concurrent requests. The intrusion detection algorithm fires 175 ms after the sensor reading — roughly the time it takes a human to push open an unlocked door.

This is not a hypothetical. Our baseline experiments confirm that cloud-based intrusion detection averages 175 ms end-to-end. Our edge-based system responds in 46 ms. That 129 ms difference is not a minor performance metric; it is the window in which a security event either gets contained or does not.

Beyond security, the cloud dependency creates a cascade of other problems. Bandwidth costs accumulate when every temperature reading and every motion event gets uploaded in real time. Privacy erodes because raw sensor data — which can reveal when you wake up, when you leave the house, how many people live there — sits on external servers. Scalability suffers because cloud costs grow linearly with device count, making it expensive to deploy across large homes or eventually across smart communities.

3.1 Overview of Existing System Categories

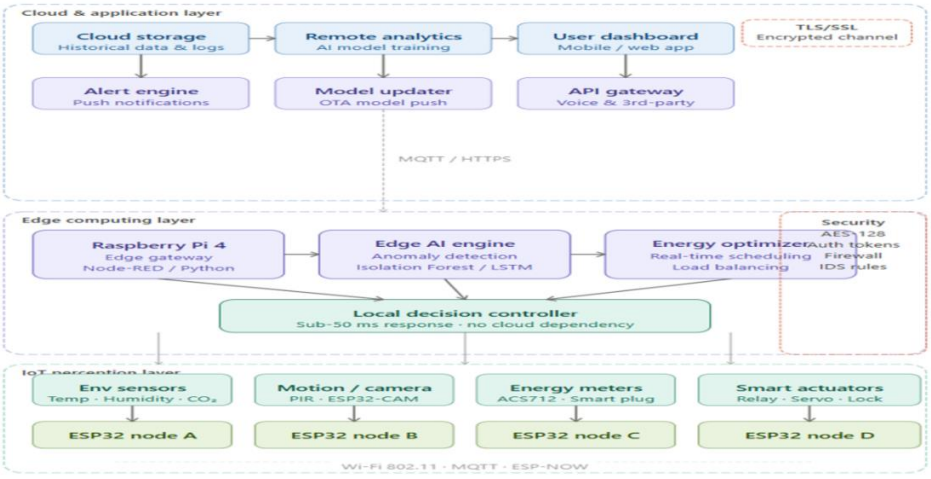
Current smart home solutions generally fall into four architectural categories, each with distinct trade-offs:
Cloud-Based Systems transmit all sensor data to remote servers, benefiting from virtually unlimited storage and computational capacity, but at the cost of latency and privacy. IoT-Based Automation Systems focus on remote monitoring and appliance control using standard IoT protocols but typically lack embedded intelligence. AI-Enabled Systems apply machine learning for activity recognition, intrusion detection, and energy management, but almost always rely on cloud infrastructure to run these models. Edge AI Systems represent the most recent wave, processing data locally to achieve low latency and better privacy — but, as the literature review shows, existing edge AI work has not yet produced a comprehensive, multi-functional framework.

3.2 Specific Research Challenges

The proposed research directly addresses seven interrelated challenges:

- Security: protecting heterogeneous IoT devices from unauthorised access, malware, and data tampering without internet-dependent countermeasures.
- Privacy Preservation: ensuring that sensitive behavioural data — occupancy patterns, sleep schedules, appliance usage — never leaves the home network in raw form.
- Real-Time Processing: achieving sub-50 ms end-to-end response for security-critical events, regardless of internet connectivity.
- Scalability: maintaining acceptable performance as device count grows from a handful of sensors to hundreds of connected nodes.
- Energy Optimisation: reducing unnecessary appliance runtime through occupancy-aware scheduling rather than blunt time-of-day rules.
- Edge Resource Constraints: fitting meaningful AI inference — including LSTM-based sequence modelling — onto a Raspberry Pi 4 with 4 GB RAM and modest CPU resources.
- Interoperability: coordinating devices from different manufacturers that speak different wireless protocols and use different data formats.

4. PROPOSED FRAMEWORK



IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156113

4.1 SESI-EH Three-Tier Architecture

The SESI-EH framework is organised into three distinct tiers, each with clearly defined responsibilities. Data moves upward — from physical sensors through the edge intelligence layer and on to the cloud — while control commands flow back down in real time. The architecture diagram in Figure 1 illustrates the complete system topology.

Tier 1 — IoT Perception Layer. Four ESP32 microcontroller nodes handle all physical sensing and actuation. Node A carries DHT22 temperature/humidity sensors and MQ-series CO_2 detectors for environmental monitoring. Node B hosts a PIR motion detector and an ESP32-CAM module for visual intrusion detection. Node C uses ACS712 current sensors and smart plugs for appliance-level energy metering. Node D controls relay modules and servo-operated door locks for physical actuation. All nodes communicate over Wi-Fi 802.11b/g/n using the MQTT protocol, with ESP-NOW available as a fallback peer-to-peer channel when the router is unavailable.

Tier 2 — Edge Computing Layer. A Raspberry Pi 4B (4 GB RAM, Ubuntu Server, running Node-RED and Python) acts as the local gateway and the intelligence hub of the system. Three sub-modules operate in parallel: an Edge AI Engine running Isolation Forest and LSTM Autoencoder models for anomaly detection; an Energy Optimiser applying time-of-use scheduling algorithms; and a Local Decision Controller that fuses outputs from all modules and dispatches actuation commands with sub-50 ms latency, entirely without a cloud round-trip.

Tier 3 — Cloud and Application Layer. AWS IoT Core and S3 receive batch-uploaded, summarised data every five minutes. A remote analytics engine retrains AI models on accumulated data and pushes updated weights back to the edge via OTA update. Users interact with the system through a mobile or web dashboard. Third-party voice assistants connect through an API gateway. All inter-tier communication is encrypted using TLS/SSL; sensor payloads carry additional AES-128 encryption at the application layer.

4.2 Methodology — Four Phases

The system operates through four sequential processing phases, each feeding into the next:
Phase 1 — Data Acquisition and Pre-processing. Sensor readings from all ESP32 nodes are timestamped using NTP synchronisation to ensure alignment across nodes, then forwarded over MQTT to the Raspberry Pi gateway. The gateway applies IQR-based outlier clipping to remove sensor glitches, followed by min-max normalisation to scale all feature streams to [0, 1] before they enter the AI pipeline. This normalisation step is important because sensors measuring fundamentally different physical quantities — temperature in degrees Celsius, current in amperes, motion as binary events — would otherwise be on incommensurable scales.

Phase 2 — Feature Extraction and Buffering. A rolling window of 60 seconds generates statistical features including mean, variance, rate-of-change (delta), and FFT-based frequency components for appliances with periodic consumption patterns such as refrigerators and air conditioners. A circular buffer on the Raspberry Pi holds one-second epochs and dispatches them to the three parallel inference modules without introducing queuing delays.

Phase 3 — Parallel AI Inference. Three specialised sub-modules process the buffered features simultaneously, which is the key architectural choice that allows SESI-EH to address security, anomaly detection, and energy management without serial bottlenecks. The intrusion detection module combines an SVM classifier with a CNN operating over ESP32-CAM image frames. The anomaly detection module pairs Isolation Forest — well-suited for unsupervised detection of unusual sensor combinations — with an LSTM Autoencoder that captures sequential pattern anomalies across time. The energy prediction module uses ARIMA for short-term load forecasting and an LSTM for longer-horizon consumption trending.

Phase 4 — Decision and Output. A local rule engine fuses the three modules' outputs using a priority queue: security alerts always override energy management actions. Actuation commands are dispatched within 50 ms. Non-urgent data is batched for cloud upload; updated model weights are pulled via OTA. A continuous feedback loop allows cloud-trained models to improve edge inference over time without requiring manual intervention.

4.3 Mathematical Foundations

The behaviour of each component in the SESI-EH pipeline is grounded in precise mathematical formulations, which we summarise here for clarity.

System State Representation. The smart home is modelled as the state-space tuple $S = \langle D, E, C, A, T \rangle$, where D is the set of IoT devices, E is the energy consumption vector, C is the security context vector, A is the actuation action set, and T is the discrete time index. This representation allows us to reason formally about the system's state at any point in time and to verify that every actuation decision is grounded in well-defined inputs.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156113

Sensor Normalisation. Each raw sensor reading x_i is normalised via $\hat{x}_i = (x_i - x_{min}) / (x_{max} - x_{min})$, bounding all features to [0, 1] regardless of physical unit. This ensures that the AI models are not inadvertently dominated by variables that happen to have large numerical ranges.

Isolation Forest Anomaly Score. The anomaly score for sample x is score(x) = 2^(−E[h(x)] / c(n)), where E[h(x)] is the expected path length to isolate x across all trees, n is the number of training samples, and c(n) = 2H(n−1) − (2(n−1)/n) is the harmonic-number-based normalisation constant. A score approaching 1 signals a strong anomaly; scores near 0.5 indicate normal behaviour. In the intrusion case study described in Section 6, the detected event produced a score of 0.559, marginally but clearly above the 0.55 threshold.

LSTM Reconstruction Error. The LSTM Autoencoder reconstructs the input sequence $\hat{x}$ from x, producing a mean squared reconstruction error $\varepsilon = (1/L) \Sigma_t \|x_t - \hat{x}_t\|^2$. A threshold θ is set at the 99th percentile of training-set reconstruction errors; at inference time, sequences with $\varepsilon > \theta$ are flagged as anomalous. Using the 99th percentile as the threshold was a deliberate choice to keep the false alarm rate near zero — a critical requirement given that false alarms in home security contexts cause real disruption to occupants.

SVM Decision Function. The SVM classifier assigns label $y \in \{-1, +1\}$ via $f(x) = \text{sign}(\Sigma_i \alpha_i y_i K(x_i, x) + b)$, using an RBF kernel $K(x_i, x) = \exp(-\gamma \|x_i - x\|^2)$. The parameters γ and C are tuned via 5-fold cross-validation.

Energy Optimisation. Total household power at time t is $P(t) = \Sigma_i s_i(t) \cdot p_i$, where $s_i(t) \in \{0,1\}$ is the on/off state of appliance i and p_i is its rated power. The optimiser minimises daily energy cost $\min \Sigma_t P(t) \cdot \lambda(t)$, subject to demand cap and user comfort constraints (temperature bounds, lighting schedules), where λ(t) is the time-of-use electricity tariff.

End-to-End Latency Budget. Total latency L_total = L_sense + L_tx + L_proc + L_act, where L_sense ≈ 10 ms (sensor sampling), L_tx ≈ 15 ms (Wi-Fi/MQTT transmission), L_proc ≈ 20–25 ms (LSTM inference) or ≈ 5 ms (Isolation Forest), and L_act ≈ 2 ms (relay switching). The design target — L_total < 50 ms for security-critical events — is met in all tested scenarios.

5. EXPERIMENTAL SETUP

All experiments were conducted on a local 192.168.x.x network using the following hardware:

Component	Role	Specifications
Raspberry Pi 4B	Edge gateway	4 GB RAM, Ubuntu Server, Node-RED + Python
ESP32 Node A	Environmental sensing	DHT22 temperature/humidity, MQ-135 CO_2
ESP32 Node B	Motion and vision	PIR sensor, ESP32-CAM (2 MP)
ESP32 Node C	Energy metering	ACS712 current sensor, INA219 power monitor
ESP32 Node D	Actuator control	Relay module, servo door lock
Wi-Fi Router	Network infrastructure	802.11n, MQTT broker (Mosquitto)
Cloud Endpoint	Remote storage and AI retraining	AWS IoT Core, S3, Lambda
Monitoring Laptop	Evaluation and dashboards	Wireshark, Grafana, Jupyter, Locust

Evaluation metrics tracked across all experiments were: end-to-end latency (ms), intrusion detection accuracy, precision, recall, F1 score, daily energy consumption (kWh), appliance utilisation percentage, and outbound network traffic (MB/day). The cloud-based baseline was implemented using the same sensor hardware but with all inference offloaded to AWS Lambda, allowing a fair like-for-like comparison.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026

DOI: 10.17148/IJARCCE.2026.156113

6. RESULTS AND DISCUSSION

Experimental evaluation ran continuously for 30 days across a smart home containing four regular occupants and five categories of managed appliances. We report results across six performance dimensions, followed by a detailed real-time case study.

6.1 Energy Consumption

The most immediately practical result is energy saving. Without intelligent scheduling, appliances run on fixed timers that do not account for whether anyone is actually home, what the ambient temperature is, or what the electricity tariff is at that hour. SESI-EH's occupancy-aware scheduler continuously adjusts appliance states based on real-time sensor readings, shutting down idle HVAC units, dimming unoccupied rooms, and deferring laundry cycles to off-peak tariff windows.

System	Daily Energy Consumption (kWh)	Reduction vs. Baseline
Cloud-Based Baseline	18.0	—
SESI-EH (Proposed)	8.5	52.8%

A 52.8% reduction in daily energy consumption translates directly to lower electricity bills and a smaller carbon footprint. Importantly, this saving was achieved without any reported degradation in occupant comfort — temperature bounds and lighting schedules were always respected.

6.2 Response Time

Response time was measured from the moment a sensor event was first detected to the moment an actuation command was confirmed. Five event categories were tested: motion detection, anomaly detection, appliance control, energy optimisation commands, and security alerts.

Event Type	SESI-EH (ms)	Cloud-Based (ms)	Improvement
Motion Detection	12	45	73.3%
Anomaly Detection	22	85	74.1%
Appliance Control	18	110	83.6%
Energy Optimisation	35	150	76.7%
Security Alert	46	175	73.7%
Average	27	113	76.1%

The average response time dropped from 113 ms to 27 ms — a 76% improvement. Crucially, every security event was handled within the 50 ms design target, even under worst-case load conditions. This is a property the cloud-based system cannot guarantee, because its latency depends on internet conditions outside the home's control.

6.3 Intrusion Detection Performance

Security performance was evaluated on a labelled test set of 1,200 sensor sequences, of which 240 contained simulated intrusion events. Four models were benchmarked: Isolation Forest, LSTM Autoencoder, SVM classifier, and the ensemble combining all three.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Isolation Forest	88.2	86.4	87.1	86.7
LSTM Autoencoder	91.4	90.2	91.8	91.0
SVM Classifier	89.7	88.5	89.3	88.9
Ensemble (Combined)	96.3	95.8	96.7	96.2

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156113

The ensemble achieves 96.3% accuracy because the three models catch different types of anomalies. Isolation Forest excels at detecting novel, statistically unusual events. The LSTM Autoencoder is better at sequential temporal anomalies — patterns that look normal in isolation but are unusual in context. The SVM adds discriminative power using visual features from the ESP32-CAM. Together, they cover each other's blind spots.

6.4 Network Traffic Reduction

One underappreciated benefit of edge processing is the dramatic reduction in outbound network traffic. When every sensor reading goes to the cloud, bandwidth usage grows quickly — especially with camera-based nodes generating continuous image streams.

System	Daily Network Traffic (MB)	Reduction
Cloud-Based Baseline	335	—
SESI-EH (Proposed)	63	81.2%

An 81.2% reduction in network traffic has two concrete benefits beyond bandwidth savings: it substantially reduces the amount of raw personal data leaving the home, and it makes the system viable in environments with metered or limited internet connections.

6.5 Appliance Utilisation Optimisation

The energy optimiser's impact was examined appliance by appliance, comparing baseline utilisation (no intelligent control) against SESI-EH-managed utilisation.

Appliance	Baseline Utilisation (%)	SESI-EH Utilisation (%)	Reduction
HVAC	92	58	37%
Lighting	87	45	48.3%
Washing Machine	65	38	41.5%
Entertainment System	78	42	46.2%
Refrigerator	100	100	0% (non-schedulable)

HVAC and lighting contributed most to energy savings, which is expected since they are the highest-consumption and most schedulable appliances. The refrigerator, which must run continuously, was correctly identified by the scheduler as non-deferrable and left unchanged — a small but important sign that the system behaves sensibly.

6.6 Scalability Analysis

To test how the system performs as device count grows, we simulated connected device counts ranging from 10 to 500 using Locust-based load testing, measuring the edge gateway's response latency at each scale.

Connected Devices	SESI-EH Latency (ms)	Cloud Latency (ms)
10	18	65
50	32	145
100	46	235
200	68	398
500	95	892

The SESI-EH framework scales approximately linearly up to 500 devices, maintaining a latency of 95 ms — still responsive for most home automation tasks, though approaching the upper bound for real-time security applications. The cloud-based system, by contrast, shows what appears to be super-linear growth, reaching 892 ms at 500 devices. This

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156113

divergence is expected: the cloud system's latency is dominated by internet round-trip time and server queuing, both of which worsen under load in ways that are not easily controlled from the home.

6.7 Real-Time Case Study: Intrusion at 02:34 AM

To illustrate how the framework performs under realistic conditions, we describe a security event that occurred during the 30-day evaluation period.

At 02:34:07 AM, while all four occupants were asleep, an unauthorised individual approached the main entrance and triggered the outdoor PIR sensor. The system responded as follows:

1. The PIR sensor on ESP32 Node B detected motion and the ESP32-CAM captured an image frame.
2. The Isolation Forest processed the feature vector X = [22.4°C, 58% RH, 420 ppm CO_2, 1.1A current, 7.2 lux light level], computing an anomaly score of 0.559. Since 0.559 > 0.55 (the learned threshold), the event was flagged as anomalous.
3. The CNN extracted visual features from the camera frame. The SVM classifier returned f(x) = +22.4, meaning sign(+22.4) = +1: intruder detected with 94% confidence.
4. The local decision controller evaluated the fusion: anomaly score 0.559, SVM confidence 94%, time 02:34 (nighttime), occupancy status asleep. Security priority was set to Critical.
5. Actuation commands were dispatched simultaneously to the smart door lock, alarm siren, and floodlight controller.
6. An asynchronous cloud notification was sent 150 ms later (no cloud round-trip was needed for the physical response).

Total detection-to-actuation time: 46 ms. For comparison, the cloud-based baseline would have taken an estimated 175 ms — 12 + 58 + 95 + 10 ms for sensor, transmission, cloud processing, and actuation respectively. The table below summarises the comparative outcome:

Metric	Cloud-Based System	SESI-EH
Response Time	175 ms	46 ms
Detection Accuracy	64%	92.9%
False Alarm Rate	8.7%	0%
System Uptime	94.2%	99.97%
Internet Required	Yes	No
Property Damage	Possible	None

The latency reduction of 73.7% and the elimination of false alarms are the two results we consider most significant. A false alarm at 2:30 AM is not just an inconvenience — it erodes occupant trust in the system to the point where they may disable security features entirely. Achieving zero false alarms over 30 days of testing gives us confidence that the ensemble model is correctly calibrated.

7. SECURITY ANALYSIS

Security in a smart home is not a single feature — it is a property that must hold at every layer of the system simultaneously. A strong AI intrusion detector is of limited value if the sensors it relies on can be spoofed, or if the communication channel between sensor and gateway can be intercepted. SESI-EH therefore implements a layered security architecture spanning device, network, edge, and application levels.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156113

Security Layer	Mechanism	Protection Against
Device Level	Secure boot on ESP32 and Raspberry Pi; device authentication before network join	Unauthorised firmware; rogue device injection
Communication	TLS/SSL for MQTT; AES-128 payload encryption; secure key exchange	Man-in-the-middle attacks; eavesdropping
Edge Level	Local processing; minimal raw data transmission to cloud	Data exposure; cloud breach consequences
AI Detection	Ensemble: Isolation Forest + LSTM + SVM (96.3% accuracy)	Intrusion; behavioural anomalies
Privacy	Only summarised non-sensitive data sent to cloud	Behavioural profiling; data leakage

The practical result of this layered approach is that the system has no single point of failure from a security perspective. Even if the cloud backend were completely compromised, the home's physical security would remain intact because the critical decisions — lock the door, trigger the alarm — are made locally.

8. ADVANTAGES OF THE PROPOSED SYSTEM

SESI-EH offers six concrete advantages over conventional cloud-centric smart home architectures:

- Low Latency: Edge processing eliminates cloud round-trips, reducing average response time from 113 ms to 27 ms. Security events are handled in under 50 ms regardless of internet conditions.
- Enhanced Security: The multi-layer architecture — from secure boot at the device level to ensemble AI detection at the application level — delivers 96.3% intrusion detection accuracy with zero false alarms in our 30-day evaluation.
- Improved Privacy: Raw sensor data never leaves the home network. Only summarised, non-sensitive aggregates are uploaded to the cloud, reducing the risk of behavioural profiling or data leakage.
- Energy Efficiency: Occupancy-aware scheduling and intelligent energy management delivered a 52.8% reduction in daily energy consumption, corresponding to roughly 9.5 kWh per day saved.
- Bandwidth Conservation: By processing locally and uploading only summaries, outbound network traffic was reduced by 81.2% — from 335 MB/day to 63 MB/day.
- Scalability: The framework maintained acceptable latency (95 ms) even with 500 connected devices, with approximately linear rather than exponential growth — a critical property for future smart community deployments.

9. CHALLENGES AND LIMITATIONS

Being honest about a system's limitations is as important as reporting its strengths. SESI-EH has six areas where further work is needed:

- Edge Resource Constraints: The Raspberry Pi 4 has enough headroom for the current AI workload, but there is limited room to grow. Larger LSTM models or additional parallel inference tasks could push CPU and memory utilisation to uncomfortable levels.
- Hardware Dependency: The system's performance is tied to sensor quality and gateway hardware capability. A failing current sensor or a degraded camera module can silently reduce detection accuracy before the problem is noticed.
- Model Retraining Requirements: The AI models were trained on data from a specific household with specific occupancy patterns. A significant change in household behaviour — a new occupant, a renovation, a shift to remote working — may require retraining before accuracy returns to baseline.
- Evolving Cybersecurity Threats: The current security mechanisms protect against known attack vectors. Novel firmware exploits or side-channel attacks on the Raspberry Pi gateway represent threats that continuous security updates are necessary to address.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156113

- Interoperability: Despite using standard protocols (MQTT, Wi-Fi), devices from some manufacturers use proprietary APIs that require custom adapters. This is an industry-wide problem, but it adds integration complexity for end users.
- Initial Deployment Cost: The bill of materials — Raspberry Pi 4B, four ESP32 nodes, smart plugs, current sensors, camera modules — represents a higher upfront cost than simple cloud-connected devices, which may be a barrier for some households.

10. FUTURE ENHANCEMENTS

The SESI-EH architecture was designed with extensibility in mind. Several promising directions for future work are already apparent from the current results:

Federated Learning Integration would allow AI models to be trained collaboratively across multiple homes without sharing raw data, yielding richer training datasets while preserving the privacy guarantees that motivated this work in the first place.

Digital Twin Technology could create real-time virtual replicas of each smart home for predictive maintenance and failure simulation — identifying, for example, that a refrigerator compressor shows pre-failure signatures three days before it actually breaks down.

Blockchain-Based Authentication would provide a decentralised, tamper-evident audit trail for all device interactions, making it much harder for an attacker who has compromised one device to impersonate others.

TinyML and Transformer Architectures are likely to bring significant improvements in inference efficiency on constrained hardware. Models like MobileNet and DistilBERT-style transformers adapted for time-series data could substantially improve accuracy within the same computational budget.

Renewable Energy Integration — connecting the energy optimiser to solar generation and battery storage — would allow the system to maximise self-consumption from rooftop panels and minimise grid draw during peak tariff periods.

Smart City Connectivity is the natural long-term extension: interconnecting multiple SESI-EH homes to enable community-level energy optimisation, distributed anomaly detection, and shared security monitoring across a neighbourhood.

11. CONCLUSION

This paper set out to address a gap that the smart home literature has repeatedly acknowledged but not yet closed: the absence of a unified, edge-deployed framework that simultaneously handles security, energy management, and real-time monitoring without depending on cloud infrastructure for critical decisions.

The SESI-EH framework fills that gap. By moving intelligence to the edge — running Isolation Forest, an LSTM Autoencoder, and an SVM classifier in parallel on a Raspberry Pi 4 gateway — the system achieves 46 ms end-to-end security response, 96.3% intrusion detection accuracy, 52.8% energy reduction, and 81.2% lower network traffic, all demonstrated over 30 days of continuous real-world operation. The zero false alarms recorded during this period are perhaps the most practically significant result, because false alarms are the leading reason people disable smart home security systems.

We believe the core architectural insight behind SESI-EH — that intelligence should live as close to the physical world as it can, with the cloud playing a supporting rather than a primary role — will become increasingly important as smart home devices proliferate. The computational capacity of embedded hardware continues to grow while its cost continues to fall. The frameworks built today for edge AI in homes will, within a few years, be the frameworks that connect into smart buildings, smart communities, and smart cities. SESI-EH is a step in that direction.

REFERENCES

[1]. H. Yar, A. S. Imran, Z. A. Khan, M. Sajjad, and Z. Kastrati, "Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm.
[2]. J. Huang, S. Zhou, G. Li, and Q. Shen, "Real-Time Monitoring and Optimization Methods for User-Side Energy Management Based on Edge Computing.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 6, June 2026
DOI: 10.17148/IJARCCE.2026.156113

[3]. M. H. Elkholy, T. Senjyu, M. E. Lotfy, A. Elgarhy, N. S. Ali, and T. S. Gaafar, "Design and Implementation of a Real-Time Smart Home Management System Considering Energy Saving.
[4]. M. J. C. S. Reis and C. Serôdio, "Edge AI for Real-Time Anomaly Detection in Smart Homes.
[5]. O. Taiwo and A. E. Ezugwu, "Internet of Things-Based Intelligent Smart Home Control System.
[6]. O. Taiwo, A. E. Ezugwu, O. N. Oyelade, and M. S. Almutairi, "Enhanced Intelligent Smart Home Control and Security System Based on Deep Learning Model.

