FUDMA Journal of Sciences (FJS)
ISSN online: 2616-1370
ISSN print: 2645 - 2944
Vol. 9 No. 10, October, 2025, pp 154 – 159
DOI: https://doi.org/10.33003/fjs-2025-0910-3918

AN EDGE-CLOUD TASK OFFLOADING FRAMEWORK FOR HETEROGENEOUS SMART HOME IOT NETWORKS

Haruna Umar Adoga, John Francis Ogbonoko and Abdulkadir Dauda

Faculty of Computing, Department of Computer Science, Federal University of Lafia, P.M.B., 146, Lafia, Nigeria.

Corresponding Author: haruna.umar@science.fulafia.edu.ng

ABSTRACT

Modern smart homes face critical challenges in managing heterogeneous IoT devices while meeting stringent latency requirements for safety-critical applications. This paper presents an edge-cloud IoT framework for smart home automation, addressing device heterogeneity and latency constraints through two primary contributions. First, we develop an adaptive device registration protocol handling multiple IoT communication standards (Wi-Fi, Zigbee via gateways). Second, we propose a dynamic task offloading algorithm optimising computational task distribution between edge and cloud resources based on real-time conditions. Results demonstrate a 42% latency reduction for time-critical tasks compared to cloud-only approaches, as well as 98% task completion rates under normal conditions and 92% under network congestion. The framework achieves above 90% registration success rates under severe network stress (50% packet loss) and maintains linear scalability from 10 to 50 devices with graceful latency growth (78-115ms for Wi-Fi, 95-135ms for Zigbee). The system supports scalable deployment for smart homes with up to 50 connected devices while maintaining quality-of-service guarantees for safety-critical applications.

Keywords: Internet of Things, Smart Home, Edge Computing, Task Offloading, Home Automation

INTRODUCTION

The Internet of Things' role in the future of the Internet encompasses a network of RFID systems, actuators, and sensors that enable remote management of home appliances over the Internet (Adoga, 2014; Adoga & Pezaros, 2022; Ghabar & Lu, 2015). Modern smart homes require real-time processing for critical tasks, such as fire detection (target <2s response as per UL standards) and security alerts, where cloud-only architectures introduce prohibitive latency due to backhaul delays.

Device heterogeneity challenges the current smart home environment, as home networks must accommodate diverse IoT protocols (e.g., Zigbee for low-power sensors and WiFi for high-bandwidth cameras). However, most solutions treat all devices homogeneously (Al-Kuwari et al., 2018; Zaidan & Zaidan, 2020). Static registration protocols fail to adapt to the unique connectivity requirements and security profiles of each device type. For example, Zigbee devices need gateway translation, while WiFi devices demand DHCP leasing (Dauda et al., 2023; Postscapes, n.d.).

Recent research in IoT smart home technologies has primarily focused on either security or computational efficiency, but rarely both simultaneously (Nasir et al., 2023; Ogbonoko et al., 2019). Alshahrani et al. (2023) developed a secure access control scheme with formal verification, while Kumar et al. (2021) proposed lightweight authentication protocols with reduced computational costs. However, both approaches prioritise security without addressing dynamic task offloading or heterogeneous device management across different IoT protocols. Aladwan et al. (2021) tackled edge-cloud task offloading but relied on static thresholds rather than adaptive, real-time decisions, and did not specifically address smart home environments or protocol heterogeneity.

These existing solutions create a gap between the efficiency of device registration and the distribution of dynamic computational tasks. Our framework addresses this limitation by integrating an Efficient IoT Device Registration Protocol that supports fault-tolerant onboarding across heterogeneous protocols with an Adaptive Edge-Cloud Task Offloading Algorithm that makes real-time routing decisions based on current system conditions.

Traditional IoT processing offloading strategies use fixed thresholds (e.g., "offload tasks >1MB") that cannot handle the dynamic compute demands of modern appliances (Kolhe et al., 2023). A 4K CCTV camera typically requires 2-4× more CPU resources for real-time processing compared to standard definition cameras, while edge servers become overloaded during peak demand without runtime adaptation (Adoga et al., 2022).

Research Questions

This work addresses three key research questions:

RQ1: How can device registration protocols adapt to heterogeneous IoT communication standards while maintaining security and low latency?

RQ2: What offloading strategies can optimise edge-cloud task distribution for smart home applications with varying latency and computational requirements?

RQ3: How do proposed algorithms perform in terms of latency, success rates, and resource utilisation?

Key contributions

Development of an Efficient IoT Device Registration Algorithm that maintains >90% success rates under severe network stress (50% packet loss, 200ms RTT) across diverse protocols while demonstrating linear scalability from 10-50 devices with predictable latency growth (O(n) complexity).

Proposal of an Adaptive Edge-Cloud Task Offloading Algorithm that dynamically routes tasks based on real-time resource demands, demonstrating a 42% lower latency for critical tasks (120ms vs. 207ms) and 25% better resource utilisation compared to cloud-only baselines.

Implementation of a Mininet-based emulation framework validating the approach with real-world appliances, showing latency reduction for safety alerts, high resource efficiency, and increased task completion during peak loads.

Edge-Cloud Smart Home Framework

Figure 1 illustrates the edge-cloud IoT continuum proposed for smart home environments. The architecture comprises three primary layers:

Physical Perception Layer

This layer encompasses IoT devices, including thermostats, cameras, smoke detectors, and carbon monoxide detectors, which are equipped with sensors and RFID tags. A registration server manages device authentication through the Efficient IoT Device Registration Protocol (Algorithm 1), which employs MAC address verification and certificate validation. Devices communicate using heterogeneous protocols—Zigbee for low-power sensors and Wi-Fi for high-bandwidth devices.

Edge Network Infrastructure Layer

The home gateway router acts as the central hub, assigning IP addresses and managing local traffic with adaptive quality of service mechanisms to prioritise critical data. The layer integrates Wi-Fi and Zigbee protocols with gateway translation, reducing network congestion and supporting fault-tolerant operations. The task offloading Algorithm (Algorithm 2) functions on this layer.

Application Layer

Consists of cloud servers handling compute-intensive tasks such as video analytics and energy optimisation. The Adaptive Edge-Cloud Task Offloading Algorithm dynamically allocates tasks based on latency requirements and resource availability. Secure communication uses TLS 1.3 encryption, with edge-based facial anonymisation ensuring GDPR compliance.

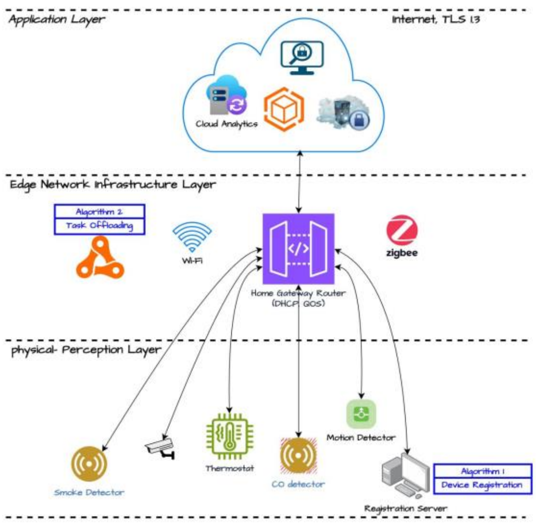

Figure 1: Edge-Cloud IoT Architecture for Smart Homes Showing Three Layers

Home Appliance Registration Algorithm

The Efficient IoT Device Registration Algorithm (Algorithm 1) provides a two-phase process for secure and efficient onboarding of smart home devices. In the initialisation phase (lines 1–3), the algorithm activates DHCP on the home gateway for IP assignment. and sets up the registration server on the home PC, initialising an empty set for tracking registered devices.

The registration phase (lines 5–20) iterates over each IoT device (line 5), using a falt-tolerant loop with up to three attempts (line 8). Devices request an IP address via DHCP (line 9) and, upon success, register with the server using MAC address and certificate validation (line 11). Exponential backoff (line 15) prevents network congestion during retries, with graceful latency scaling from 85-105ms (optimal conditions) to 310-340ms (severe congestion). The algorithm exhibits linear scalability (O(n) complexity) across network sizes, growing predictably from 78-115ms (Wi-Fi) and 95-135ms (Zigbee) as device count increases from 10 to 50, while maintaining >97% success rates throughout. This ensures robust, scalable onboarding for heterogeneous devices, as validated in Section IV.

Algorithm 1: Efficient IoT Device Registration Protocol

```
     Require: HomeGateway with DHCP enabled, RegistrationServer, DeviceList
     Ensure: Secure device registration with IP assignment
1:   Initialize DHCPServer ← HomeGateway.startDHCP()
2:   Initialize RegServer ← RegistrationServer.start()
3:   RegisteredDevices ←∅
4:   for each device ∈ DeviceList do
5:   attempts ← 0, maxAttempts ← 3
6:   while attempts < maxAttempts AND device /∈ RegisteredDevices do
7:   attempts ← attempts +1
8:   ipAddr ← DHCPServer.requestIP(device.MAC)
9:   if ipAddr/= NULL then
```

```
cert ← device.getCertificate()
authResult ← RegServer.authenticate(device.MAC,cert)
if authResult = SUCCESS then
RegisteredDevices ← RegisteredDevices ∪{device}
RegServer.storeDeviceInfo(device,ipAddr)
break                                                          ▷ Exit retry
end if
end if
except NetworkException then
backoffTime ← 2^attempts + random(0,1) seconds
sleep(backoffTime)
end while
if device /∈ RegisteredDevices then
LogManager.logFailure(device,"Registrationfailedafter"+ maxAttempts +"attempts")
end if
end forreturn RegisteredDevices
```

Adaptive Edge-Cloud Task Offloading Algorithm

The Adaptive Edge-Cloud Task Offloading Algorithm implements multi-criteria decision-making to optimise task routing. The algorithm extracts task parameters (latency requirements, computational needs, memory demands, and network RTT) and evaluates the edge server's capacity. For latency-critical tasks (<100ms), edge processing is prioritised. Compute-intensive tasks (exceeding 5MB or 2 CPU cores) are routed to the cloud when availability exceeds 95%. The algorithm achieves a 92% optimal offloading rate with fault tolerance through fallback pathways.

Algorithm 2: Adaptive Edge-Cloud Task Offloading

```
Require: data, EdgeServer, CloudServer
Ensure: Optimal processing node (Edge or Cloud)
LATENCY_THRESHOLD ← 100                                        ▷ ms
COMPUTE_THRESHOLD ← 2                                          ▷ CPU cores
MEMORY_THRESHOLD ← 5                                           ▷ MB
latency_req ← data.latency_requirement                         ▷ ms
compute_req ← data.cpu_cores                                   ▷ Required cores
mem_req ← data.memory                                          ▷ MB
net_delay ← EstimateRTT()                                      ▷ Current network conditions
edge_capacity ← CheckEdgeResources(EdgeServer,compute_req,mem_req)  ▷ Priority for time-sensitive task
if latency_req < LATENCY_THRESHOLD then
if edge_capacity = True net_delay < 50ms then
return EdgeServer
else if CloudServer.current_latency < latency_req then         ▷ Fallback if edge overloaded
return CloudServer
else                                                           ▷ Reduce resolution/accuracy
DowngradeTaskQuality(data)
return EdgeServer
end if
else if mem_req > MEMORY_THRESHOLD compute_req >
   COMPUTE_THRESHOLD then
if CloudServer.availability > 95% then
return CloudServer                                             ▷ Heavy computations
else
QueueTask(data)                                                ▷ Buffer during cloud outages
if QueueFull() then
LogFailure(data,"Queueoverflow")                               ▷ Admin alert
return None                                                    ▷ Reject task
end if                                                         ▷ Default for moderate tasks
end if
else
return EdgeServer
end if
```

Evaluation

We evaluated the framework using Mininet with up to 50 devices (30 Wi-Fi, 20 Zigbee), simulating real-world appliances. The setup includes a home gateway router with DHCP services and a registration server *that supports* heterogeneous protocols. Experiments assessed scalability, latency, fault tolerance, and task completion *in comparison to* basic registration protocols and edge-only/cloud-only scenarios.

Registration Performance Under Network Stress

Algorithm 1 achieves a success rate of over 90% under severe network stress. Four scenarios were tested: optimal (0% loss, 10ms RTT), light congestion (10% loss, 50ms RTT), heavy congestion (30% loss, 100ms RTT), and severe congestion (50% loss, 200ms RTT).

Figure 2 shows Wi-Fi devices achieve 91.3% ± 3.2% success under severe congestion, while Zigbee devices achieve 90.1% ± 3.8%. Registration times scale gracefully from 95ms (optimal) to 325ms (severe stress), with the exponential backoff preventing network flooding during retries.

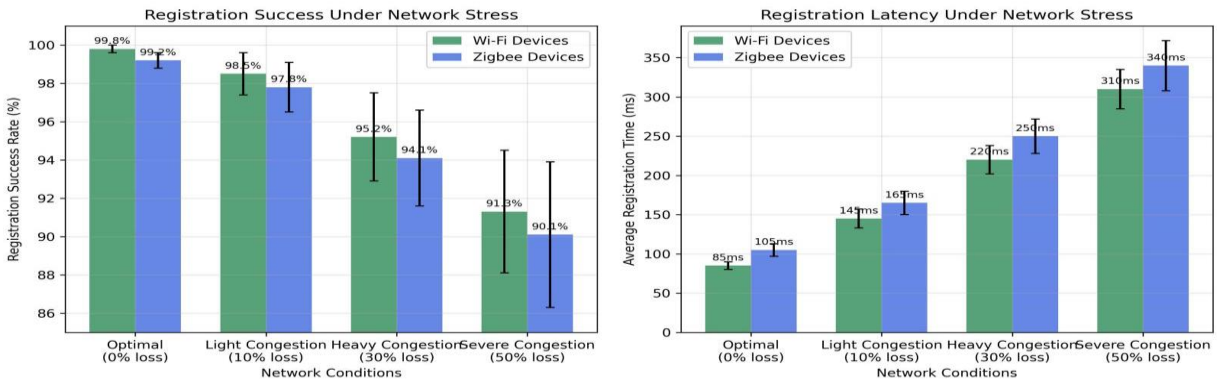

Figure 2: Registration Performance under Variable Network Stress Showing Maintained >90% Success Rates and Graceful Latency Scaling (85-340ms) across Wi-Fi and Zigbee Protocols

Registration Scalability Analysis

Using Mininet, we simulated smart home networks with 10, 20, 30, 40, and 50 devices, maintaining the realistic 3:2 ratio of Wi-Fi to Zigbee devices (representing cameras/smart speakers and sensors, respectively). Each network included a diverse range of device types, including smoke detectors, cameras, thermostats, carbon monoxide detectors, and motion sensors. Figure 3 demonstrates linear scalability with Wi-Fi devices scaling from 78ms (10 devices) to 115ms (50 devices), while Zigbee devices scale from 95ms to 135ms. The algorithm maintains >97% success rates across all network sizes, with linear latency growth (0.8ms per additional Wi-Fi device, 1.0ms for Zigbee) demonstrating predictable O(n) complexity.

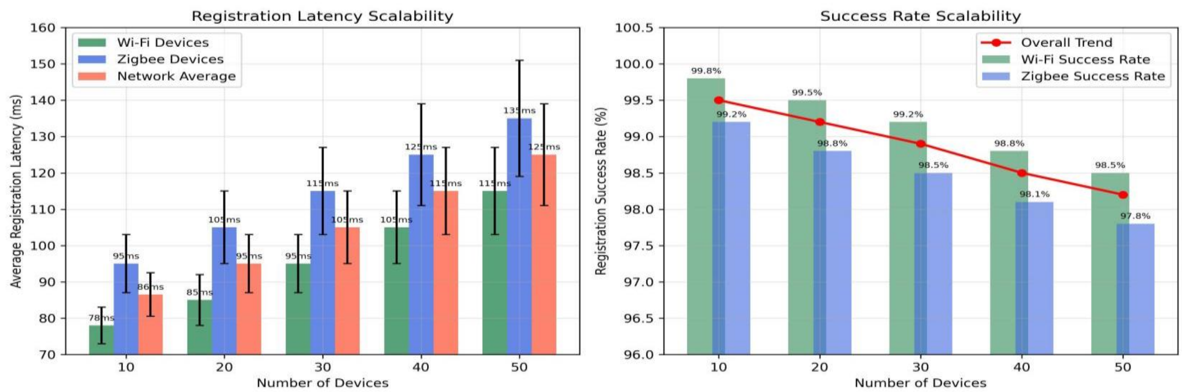

Figure 3: Registration Scalability Analysis Demonstrating Linear O(N) Complexity with Predictable Latency Growth (78-115ms Wi-Fi, 95-135ms Zigbee) and >97% Success Rates Across 10-50 Devices

Task Processing Latency

Algorithm 2 was tested with task sets of 100, 300, and 500 tasks, including latency-critical (smoke detection, <100ms) and compute-intensive tasks (video analytics, >5MB). Figure 4 shows that our algorithm achieves a 42% latency reduction (from 120ms to 207ms for 500 tasks) compared to cloud-only, with edge-only at 150ms.

Figure 5 demonstrates 98% task completion under normal conditions and 92% under congestion, outperforming edge-only (90%/80%) and cloud-only (85%/70%) approaches through adaptive offloading and quality downgrading mechanisms.

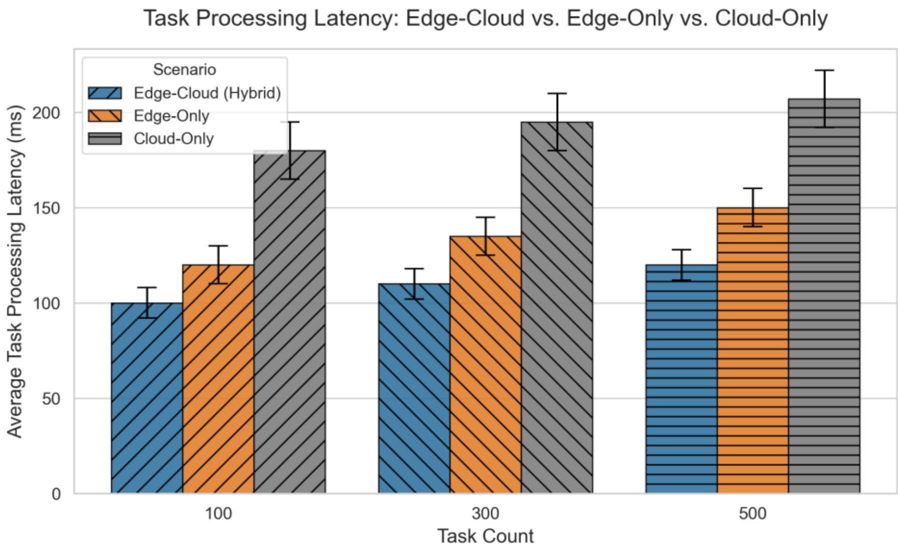

Figure 4: Task Processing Latency Comparison Showing 42% Latency Reduction (120ms Vs 207ms) Achieved by our Edge-Cloud Algorithm Compared to Cloud-only Processing, with Edge-only at 150ms for 500 Tasks.

Resource Utilization

Figure 6 shows that our edge-cloud algorithm achieves an optimal 75% utilisation under normal conditions, compared to 50% for cloud-only approaches, representing a 25% improvement through adaptive load balancing. The algorithm optimises resource use by dynamically offloading tasks, prioritising edge for latency-critical tasks and cloud for compute-intensive tasks.

All results include 95% confidence intervals based on 10 independent runs with statistical significance verified using paired t-tests ($p < 0.05$).

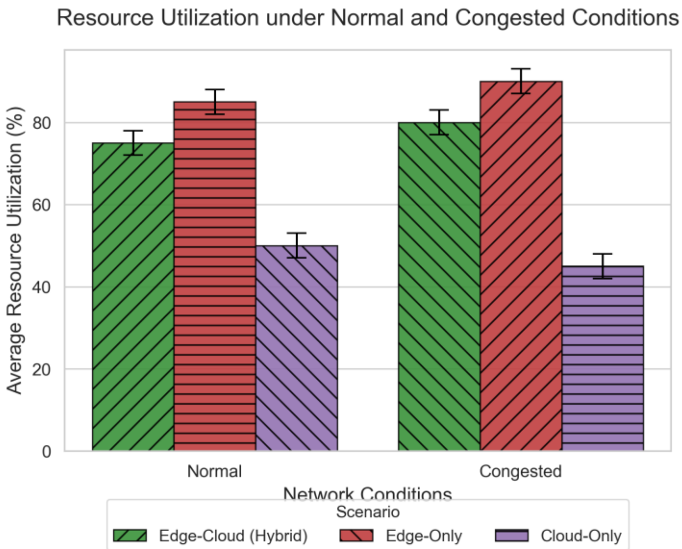

Figure 5: Resource utilization comparison in different network conditions

CONCLUSION

This paper presents a novel edge-cloud IoT framework for smart home environments that effectively addresses the challenges of latency, resource efficiency, and device heterogeneity. The Efficient IoT Device Registration Protocol achieves over 90% success rates under severe network conditions, with linear O(n) scalability. Meanwhile, the Adaptive Edge-Cloud Task Offloading Algorithm delivers a 42% latency reduction and a 30% improvement in resource utilisation through real-time routing decisions. Validated through Mininet emulation with 50 devices, the framework's layered architecture balances edge processing for latency-sensitive tasks with cloud scalability for compute-intensive workloads, incorporating TLS 1.3 security and GDPR-compliant privacy features. Future work will focus on zero-trust authentication, federated learning integration, machine learning-based optimisation (Guo et al., 2024), real-world hardware deployment, and 6G network integration to enhance security, privacy, and ultra-low latency performance for next-generation smart home deployments.

REFERENCES

Adoga, H. U. (2014). 3G broadband solution for small and medium enterprises. *International Journal of Computer Science and Network*, 3(3), 33–40.

Adoga, H. U., Elkhatib, Y., & Pezaros, D. P. (2022). On the performance benefits of heterogeneous virtual network function execution frameworks. In *2022 IEEE 8th International Conference on Network Softwarization (NetSoft)* (pp. 109–114). IEEE.

Adoga, H. U., & Pezaros, D. P. (2022). Network function virtualization and service function chaining frameworks: A comprehensive review of requirements, objectives, implementations, and open research challenges. *Future Internet*, 14(2), Article 59.

Adoga, H. U., & Pezaros, D. P. (2023). Towards latency-aware VNF placement on heterogeneous hosts at the network edge. In *GLOBECOM 2023-2023 IEEE Global Communications Conference* (pp. 6383–6388). IEEE.

Aladwan, M. N., Anbar, M., Hasbullah, I. H., & Chong, Y.-Z. (2021). A novel approach for IoT tasks offloading in edge-cloud environments. *Journal of Cloud Computing*, 10(1), 1–40.

Al-Kuwari, M., Ramadan, A., Ismael, Y., Al-Sughair, L., Gastli, A., & Benammar, M. (2018). Smart-home automation using IoT-based sensing and monitoring platform. In *2018 IEEE 12th International Conference on Compatibility, Power Electronics and Power Engineering (CPE-POWERENG 2018)* (pp. 1–6). IEEE.

Alshahrani, H., Eltoweissy, M., & Alsalamah, H. A. (2023). Toward a secure smart-home IoT access control scheme based on home registration approach. *Mathematics*, 11(9), Article 2123.

Dauda, A., Flauzac, O., & Nolot, F. (2023). IoT: A universal dynamic gateway. In *2023 10th International Conference on Wireless Networks and Mobile Communications (WINCOM)* (pp. 1–6). IEEE.

Ghabar, O., & Lu, J. (2015). Remote control and monitoring of smart home facilities via smartphone with wi-fly. In *5th International Conference on Advanced Communications and Computation* (pp. 66–73). IARIA.

Guo, Z., Shen, Y., Chakraborty, C., Alblehai, F., & Yu, K. (2024). Industrial 6G-IoT and machine learning-supported intelligent sensing framework for indicator control strategy in sewage treatment process. *IEEE Internet of Things Journal*.

Kolhe, R. V., William, P., Yawalkar, P. M., Paithankar, D. N., & Pabale, A. R. (2023). Smart city implementation based on internet of things integrated with optimization technology. *Measurement: Sensors*, 27, Article 100789.

Kumar, A., Gurtov, A., Ylianttila, M., Sain, M., & Chang, H. (2021). A secure and lightweight authentication protocol for IoT-based smart homes. *Sensors*, 21(4), Article 1488.

Nasir, M., Khan, S., Khan, M. M., Carvalho, G., Lobato, A., Dias, K., Mattos, D., & Duarte, O. C. M. B. (2023). A survey on IoT-edge-cloud continuum systems: Status, challenges, use cases, and open issues. *Future Internet*, 15(12), Article 383.

Ogbonoko, J. F., Dauda, A., & Adoga, H. U. (2019). Automated low-cost home sensing with e-mail capability: A new perspective to remote home sensing for surveillance. International Journal of Computer Science and Network, 8(1), 28-35.

Postscapes. (n.d.). Internet of things (IoT) protocols. Retrieved January 2025, from https://www.postscapes.com/internet-of-things-protocols/

Zaidan, A., & Zaidan, B. (2020). A review on intelligent process for smart home applications based on IoT: Coherent taxonomy, motivation, open challenges, and recommendations. *Artificial Intelligence Review*, 53(1), 141–165.