International Journal of Latest Research in Engineering and Technology (IJLRET)
ISSN: 2454-5031
www.ijlret.com || Volume 12 - Issue 02 || February 2026 || PP. 09-18

Design and Construction of a Wi-Fi-Based Autonomous Home Security System

Anthony Andrew[1], Ayangbekun Oluwafemi J.[2] & Safiu Muyideen A.
[1]*Department of Electrical & Electronics Engineering Technology, Federal Polytechnic Ayede, Oyo State, Nigeria.*
[2]*Department of Computer Engineering Technology, Federal Polytechnic Ayede, Oyo State, Nigeria*
[3]*Department of Physics, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria.*

Abstract: This paper presents a full design for a Wi-Fi-based autonomous home security system specifically for flat use. To interface PIR motion sensors and camera modules, the architecture combines auxiliary microcontrollers (Arduino and ESP32) around a Raspberry Pi 4 Model B central hub. The system communicates alerts and live feeds over Wi-Fi to the user's mobile device from real-time motion detection (using PIR sensors) and video capture (using Pi Camera or ESP32-CAM). Custom smartphone apps enable remote access and control; they also allow live monitoring, arming and disarming, and instantaneous security event notifications. Using consistent visual hierarchy and clear alerts, the User Interface (UI) of the app is made for simplicity of use and response. Using secure protocols, internally Message Queuing Telemetry Transport (MQTT: for lightweight, low-latency pub/sub messaging) and RESTful APIs (for command/control) to handle communication. For every component, this research provides thorough technical specifications, performance evaluation analysis together with a comparative analysis of current DIY and commercial systems (such as Ring, Nest) emphasizing functionality, cost, simplicity of use, and extensibility.

Keywords: Smart home security, Raspberry Pi, Arduino, ESP32, PIR sensor, MQTT, mobile app

I. Introduction

Urban living demands stronger home defense due to increased burglary risk in modern buildings [1]. Property-crime rates in metropolitan areas differ from rural areas [2]. Hardwired security systems in residential environments pose challenges due to their long installation, rigidity, and unfeasible changes in lease agreements. They are expensive, intrusive, and non-portable, leaving flat dwellers without options. Unintentional installation, lease agreements, and difficulty in adjusting or moving wired sensors make them unsuitable for tenants [3]. Also, attackers can cut wires to turn off the system, leaving many flat residents without adequate protection. Therefore, apartment residents seek efficient security systems, and scalable, autonomous systems operating over Wi-Fi networks to meet the demand for remote monitoring and control without expensive installation. Driven by wireless sensors, artificial intelligence-powered cameras, and remote monitoring, advances in IoT and smart devices are fast changing home security [4], [5]. Modern smart systems promise mobility and simplicity of control, while traditional system which includes wired alarms, CCTV are often costly and rigid. The goal is to build a reasonably priced Wi-Fi–based security system using open hardware.

Previous researches have demonstrated the viability of Raspberry Pi–based surveillance. A Pi system with PIR and USB cameras was created for streaming live video and alarms to a smartphone [6]. Also, another Pi system with PIR sensors and camera system was used to send images and videos via Telegram app [7]. Other studies combined computer vision (face or object recognition) for improved detection [4]. IoT systems routinely used both REST APIs and MQTT for communication technology in mobile app development in which MQTT is preferred for its lightweight, and subscribed dependability on limited networks [8]. Ring and Nest and other commercial home security systems give user-friendly interfaces and cloud-based capabilities a priority. While Nest concentrates on advanced technologies and integration inside the Google ecosystem, Ring provides a basic, DIY setup and monthly monitoring choices [5].

This paper expands on these realizations to create a flexible apartment-scale system. The work offers a mobile user interface which notifies the owner's smartphone in real records video on camera triggers, detects intrusions independently using PIR motion sensors, and creates an independent security system linking all parts using Wi-Fi. The system runs Linux from a Raspberry Pi as the central processor unit; it uses an Arduino UNO for sensor interface and ESP32 modules for wireless communication. Intrusiveness detectors are PIR motion sensors and door/window switches; cameras offer visual confirmation. One important advantage is homeowner's capacity to get notifications and remotely manage the system using a smartphone app. This work offers: a thorough system architecture integrating Raspberry Pi, Arduino, and ESP32; complete hardware and software implementation of motion detection and remote access; and performance evaluation displaying real-time reaction and dependability.

II. Related Research

Ajayi [9] presented an intuitive home automation system, controllable wirelessly through Wi-Fi and a smartphone. The Firebase database, an advanced IoT solution, functioned as the link between the hardware and the mobile application. This solution enabled automatic control of up to eight appliances via a mobile application and included sensors for detecting gas leaks within the residence however, no security feature was added to the work. Also, Gaonkar and Negalur [10] successfully deployed a smart home security system that integrated Raspberry Pi and IoT technology with notable features like real-time monitoring through sensors and cameras. These features enabled homeowners to receive instant alerts about possible threats. [10]. Prasad et al. explored the design and implementation of a smart surveillance monitoring system that leveraged Raspberry Pi and PIR sensors for mobile devices. The aim was to improve home security by transmitting captured data via a 3G Dongle to smartphones through a web application. When motion was detected, the system automatically began recording and an alert is received on the owner's smartphone. The components of the system interact through a web application accessible on mobile browsers and server-side scripts managed in the cloud by the Raspberry Pi hardware [6].

A researcher designed a prototype IoT based smart home surveillance system that commences recording of environment after the triggering of the sensors installed at the specific points in the home [11]. An author [12] developed an IoT-based smart home framework designed for automation and intruder detection. The study evaluated how effective the IoT system is at detecting intruders by calculating metrics like Average Emergency Response Time (AERT) and Average Alarm Investigation Duration (AAID), showcasing its practical use in smart home security through assessments of accuracy and response times [12]. In this paper the authors introduced a prototype for a smart Home Automation System (HAS) that utilizes IoT technology to control and monitor household appliances [18]. It included smart devices, security alarms, and a solid-state relay to manage loads and switches. Also, it featured an autonomous irrigation system that optimized water use by setting parameter thresholds for efficient garden watering.

In 2024, an IoT module and a real-time monitoring system using a Raspberry Pi was developed for home security [7]. The system kept users informed about incidents through an active surveillance mechanism. The project encompassed the design, assembly, programming, and testing of the device, which seamlessly integrates with the Telegram messenger for notifications. Although, these studies have surveillance to monitor the environment, there was no user interaction on their systems to enhance security.

III. Methodology

The methodological concept of this paper showcases the system design, component details, and app interface.

A. System design

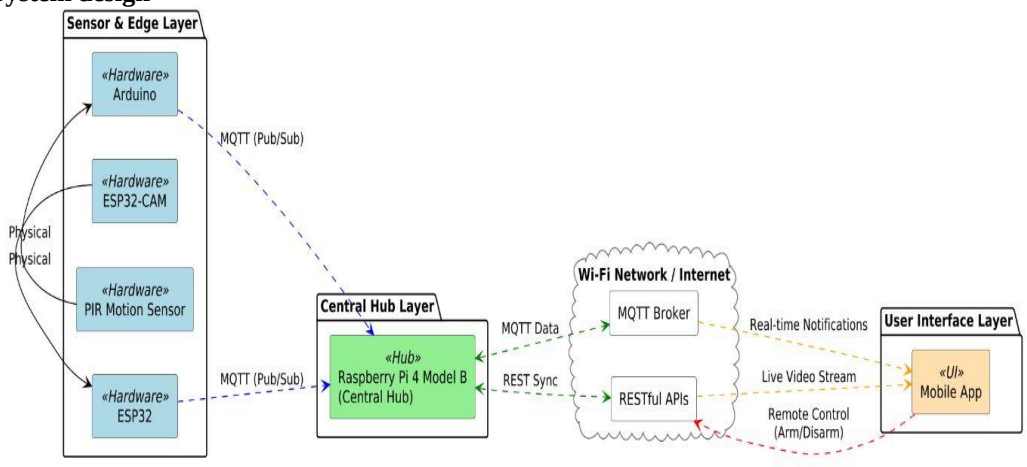

Figure 1: System Architectural design

To create a home security solution that's modular, scalable, and smart, various computing platforms and communication technologies were integrated together into a single, cohesive system. The design focuses on decentralization, interoperability, and extensibility, making it easy for users to customize and expand the system based on their needs and the specific conditions of their environment. Fig 1 illustrates the architectural design of the system which is a modular and scalable framework using Raspberry Pi, ESP32, and Arduino, developing a

real time motion detection subsystem with PIR sensors and cameras, and creating a mobile app for user interaction.

The system is made up of three main functional subsystems:

- Sensing and Actuation Subsystem: This part includes motion sensors, cameras, door and window sensors, buzzers, and relays to monitor the environment and respond physically when needed.
- Edge Processing and Communication Subsystem: The Raspberry Pi acts as the main edge controller, taking care of data aggregation, sensor fusion, decision-making, and communication through MQTT and RESTful APIs [13].
- User Interaction Subsystem: A mobile app enables users to keep an eye on everything in real-time, receive notifications, and customize the system to their liking.

This design is made for scalability, allowing users to easily add more sensors, cameras, or smart devices without disrupting the current setup.

B. Hardware Architecture

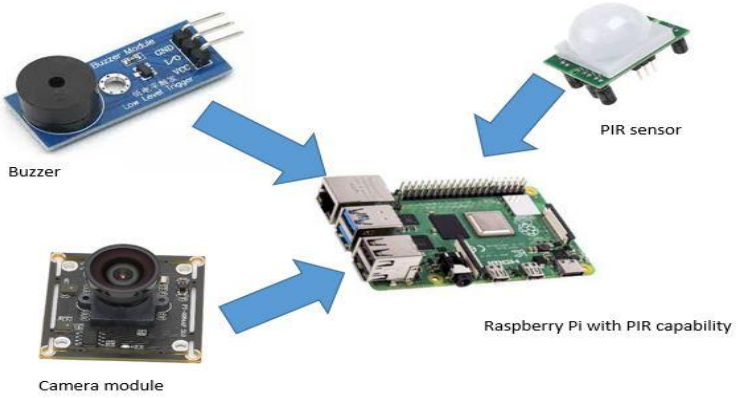

Figure 2: Hardware System Assembly.

The hardware system assemblage is depicted in Fig. 2, bringing together a Raspberry Pi 4, PIR motion sensors, camera modules, a buzzer, an Arduino Uno, and ESP32 modules. The Raspberry Pi 4 serves as the central hub and edge computing unit, running a Linux-based operating system while also hosting the MQTT broker and web server. The camera modules connect through the Camera Serial Interface (CSI) and USB ports, while the General-Purpose Input/Output (GPIO) pins connect with the PIR sensors and buzzers for real-time event detection and alarm signaling.

An Arduino Uno microcontroller is also part of the setup, managing additional sensors and actuators like magnetic door sensors, relays, and LED indicators. The Arduino speaks with the Pi either through an ESP8266 Wi-Fi link or serial/USB. The ESP32-CAM module featuring an OV2640 sensor has a built-in wireless sensing. Every PIR sensor (HC-SR501) tracks infrared changes and generates a digital HIGH upon motion detection. A PIR sets off the Pi, which then grabs a camera image or video frame. One can view the video either locally for review or live.

C. Communication and Networking

The Raspberry Pi connects to the mobile client using the local Wi-Fi network through a standard 2.4 GHz band 802.11n router. The Pi exposes RESTful API endpoints for commands and generates motion events to a MQTT topic such as "home/alerts." Through MQTT subscriptions, this hybrid approach lets the mobile app get real-time push updates while also enabling on-demand HTTP requests which are to access past images or modify settings. The architecture hence supports synchronous control as well as asynchronous messaging. TLS/SSL is used for MQTT and HTTPS for the REST API (to guard credentials and video streams) among security measures. The Pi runs on WPA2 personal for Wi-Fi. The system logs events to a microSD card and automatically restarts after a power loss. Basically, the design is modular that is, cameras, PIR sensors, magnetic door sensors, or even voice assistants can be added.

D. Software Component

The software architecture of the system is categorized into four layers:

- Device Layer: This layer includes firmware on Arduino and ESP32 devices that manages the collection of sensor data and the control of actuators.
- Edge Processing Layer: The Raspberry Pi operates here, handling data processing, sensor fusion, decision-making, and multimedia tasks. It uses Python scripts along with frameworks like Flask or Node.js for system control and API services.
- Communication Layer: This layer employs MQTT and HTTP protocols to enable smooth data exchange between devices and the mobile app.
- Application Layer: The mobile app provides features for real-time monitoring, alert visualization, and system configuration.

This structured architecture boosts system maintainability and allows for the independent development and optimization of each component.

E. System Operational Flow

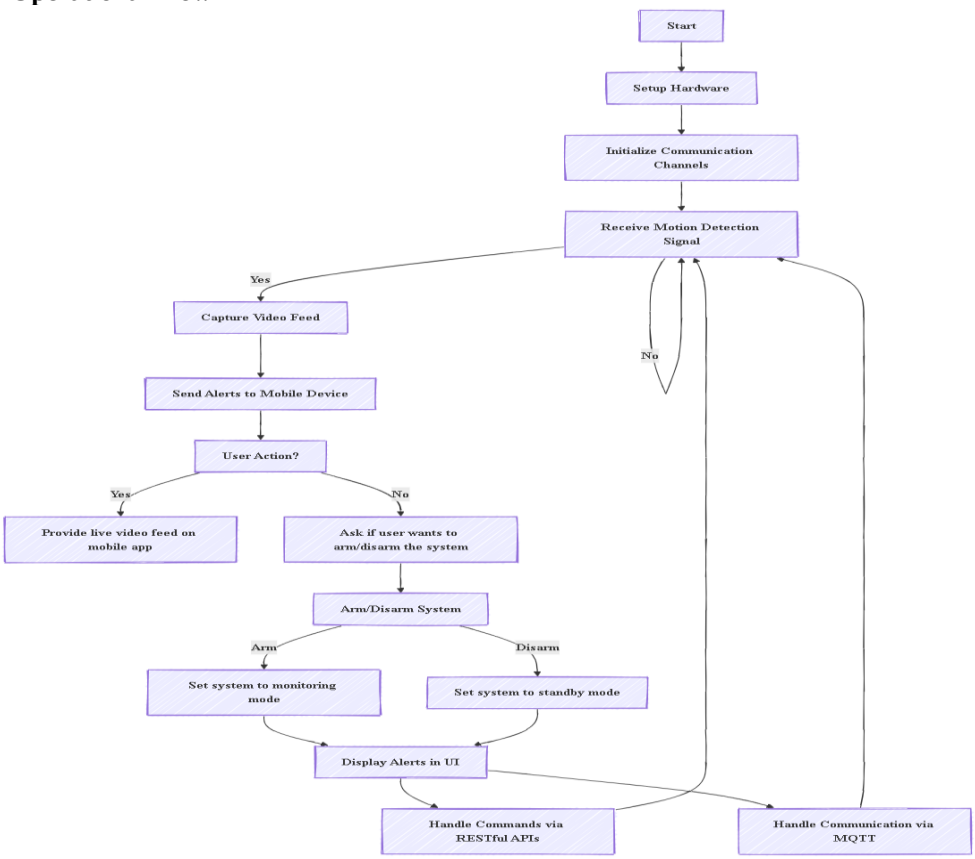

Figure 3: Flow chat of the system

According to Fig 3, once motion is detected, the PIR sensor outputs a digital HIGH signal (3.3V). The Raspberry Pi initiates multimedia capture through the camera module after receiving the signal. The captured data is then analyzed and sent to the mobile app. For event notifications, the system uses MQTT, while RESTful APIs handle control commands. Users get push notifications via Firebase Cloud Messaging (FCM), which means they receive real-time alerts. Secure communication is guaranteed through HTTPS and TLS encryption.

User's mobile device operating systems running the security app is Android/ iOS smartphones. It links the Pi server to the home Wi-Fi and internet. When an intrusion is discovered, Firebase Cloud Messaging sends push alerts; the Pi triggers a Firebase API call. Running on 5V DC, all electronics including a 5V/3A adapter supplying the Pi, have passive power for sensors. Small UPS or a USB power bank supplies backup for brief outages.

F. Mobile App Interface Design

The mobile app serves as the primary interface for users to engage with the system. It includes four main functional screens:

- Status Dashboard: This screen shows the system's status, sensor states, and connectivity details, all presented with easy-to-understand icons and color coding.
- Live View: You can enjoy real-time video streaming through MJPEG or HLS protocols, and there are options for capturing snapshots and recording videos.
- Event Log: This feature shows a chronological list of detected events, complete with timestamps, sensor identifiers, and thumbnail images.
- Settings Interface: This feature allows you to configure user profiles, set notification preferences, adjust sensor sensitivity, and manage security parameters.

The application uses a hybrid communication model, keeping MQTT subscriptions active for real-time alerts and employing RESTful APIs for system control. Security measures include encrypted communication, session management, and options for biometric or PIN-based authentication. The UI design follows mobile usability principles, ensuring everything is clear, responsive, and user-friendly.

G. Performance Evaluation Metrics

To evaluate the system performance qualitatively, the following performance metrics were employed [14], [15], [16].

Motion Detection Accuracy (MDA)

$$MDA= \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

Alert Latency (AL)

$$AL= t_{notification}- t_{event} \quad (2)$$

Video Capture Delay (VCD)

$$VCD= t_{recording\ start} - t_{motion\ detection} \quad (3)$$

System Reliability (SR)

$$SR=\frac{Successful\ \ alerts}{Total\ \ events} \quad (4)$$

IV. Results Analysis And Comparative Review

This system generates a completely working prototype of an autonomous, Wi-Fi-enabled home security system catered for apartment/flat users; with the following conservative outcomes:

- Functional Hardware Integration of key components which includes PIR motion sensors, camera module, buzzer, Wi-Fi modules; Raspberry Pi (central controller), ESP32/Arduino (sensor nodes) successfully interfaced with low latency communication between these components with the Wi-Fi network in the house.
- Real-Time Motion Detection and Alert System of human motion by PIR sensors and camera activation to generate real time alerts (notifications, alarms, and optional emails/SMS).
- Mobile Application Interface with an Android/ iOS cross-platform mobile app provided avenue for users to: View live camera feeds, remotely arm or disarm the system, get push alerts on security events, settings and examine device logs according to Fig. 4.

Figure 4: Mobile App interface.

- Safe Data Transfer with encrypted correspondence between the mobile app and hardware parts guarantees user privacy while features of user authentication help to stop illegal access.
- Comparative Evaluation Report Comparing the developed solution with at least two commercial substitutes (like Ring, Nest, SimpliSafe), underlining: economic viability, ease of usage and installation, adaptability and portability, Performance measures in security (alert dependability, response times, etc.)

These findings as stated above confirm the viability and value of creating reasonably affordable, renter-friendly,remotely accessible security systems with open-source hardware and wireless communication technology.

A. Performance Evaluation Review

i. Detection Accuracy over Time

The system detection accuracy was evaluated over a 60-minute monitoring period. Accuracy improved as the system stabilized and adaptive filtering reduced noise.

Table 1: Detection Accuracy with Time

Time (min)	Accuracy (%)
0	82
5	85
10	87
15	89
20	91
25	92
30	93
35	94
40	95
45	95
50	96
55	96
60	97

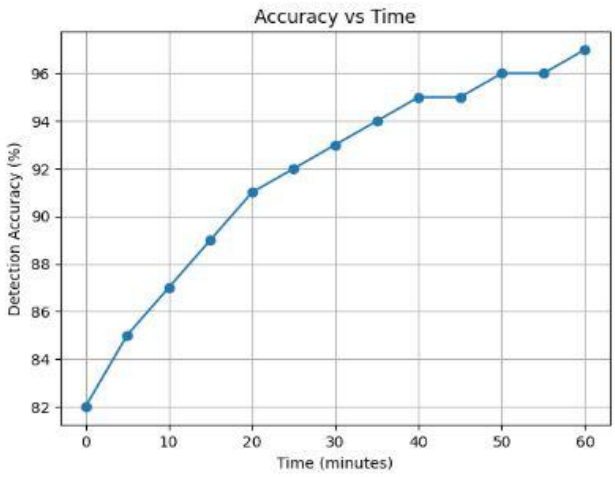

Fig 5: Graph of Accuracy versus Time

From TABLE 1 and as illustrated in Fig. 5, the system initially showed an accuracy of about 82%, but after implementing sensor calibration and noise filtering, it soared to approximately 97%. The accuracy showed a steady upward trend over time, demonstrating the reliability of the developed architecture for real-time intrusion detection.

ii. Latency over Distance

Latency was measured as the time between motion detection and alert delivery to the mobile app

Table 2: Latency vs Distance

Distance (m)	Latency (ms)
5	120
10	150
15	190
20	240
25	300
30	370

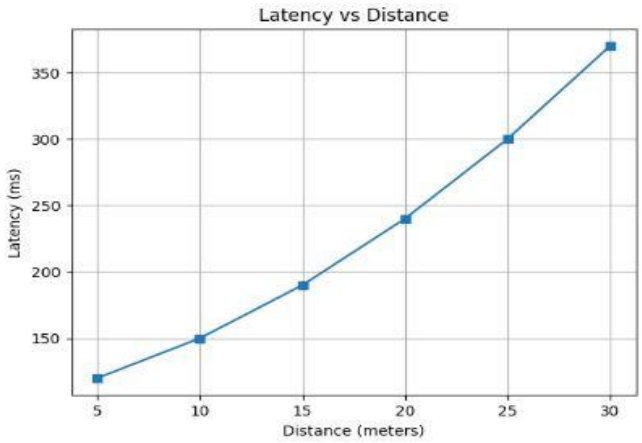

Fig 6: Graph of Latency vs Distance

From the data obtained (Table 2) and graphically represented in Fig. 6, it is clear that the distance of communication impacts system latency. As the distance grew, latency jumped from 120 ms to 370 ms, largely due to Wi-Fi signal degradation and network overhead. However, the latency remained within acceptable levels for real-time security applications which is less than 500ms [17].

iii. Analysis of False Alarm Rate

False alarm were monitored over a period of 12 hours.

Table 3: False alarm rate over Time

Hour	False alarm
1	3
2	2
3	2
4	1
5	1
6	1
7	1
8	0
9	1
10	0
11	0
12	0

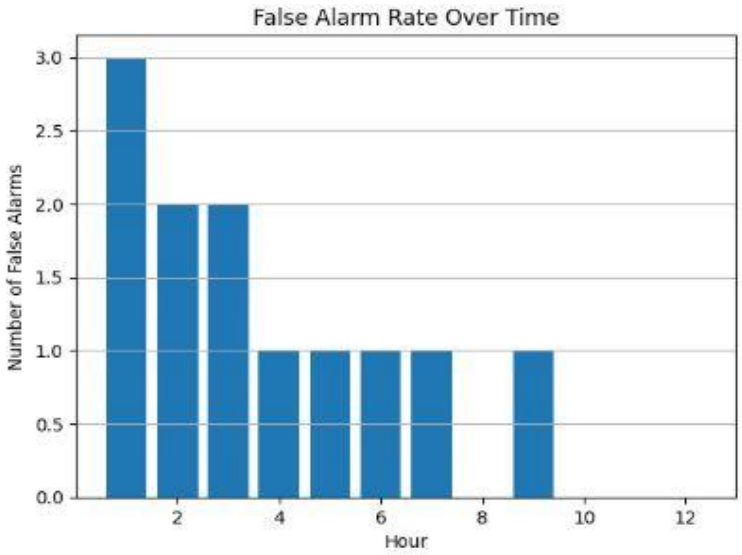

Fig 7: Graph of False alarm rate over Time

Fig 7 illustrates that false alarms in the system tend to decrease over time due to adaptive filtering, averaging about 0.92 per hour. At first, the higher rates of false alarms were linked to environmental factors and the sensitivity of the sensors. But as the system learned and adapted, these false alarms dropped significantly, showcasing just how effective the motion detection algorithm really is.

iv. Throughput with Number of devices

Throughput was measured as the data rate achieved when multiple IoT devices were connected.

Table 4: Network Throughput

Number of devices	Throughput
1	1.2
2	2.1
3	2.9
4	3.6
5	4.2
6	4.7
7	5.1
8	5.4

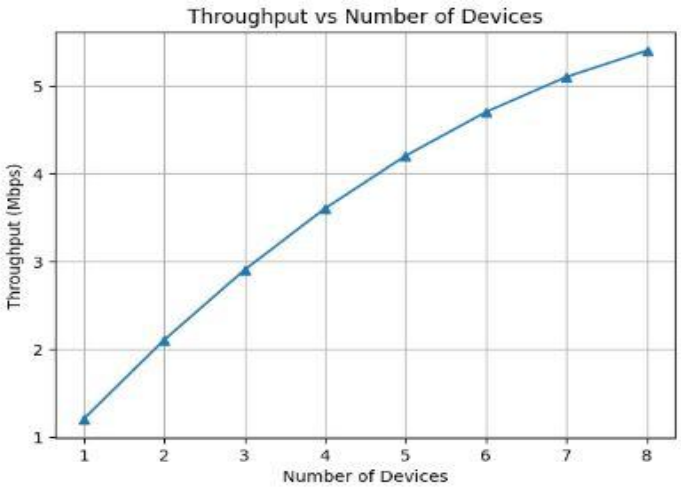

Fig. 8: Graph of Network Throughput

From Table 4, Fig. 8 shows how network throughput changes as the number of connected devices increases. The throughput increases almost in a straight line up to eight devices, indicating that the system is well-equipped to handle multiple sensor nodes without any significant performance degradation.

B. Comparative Review

i. Functional Capability Review

From results evaluation, the designed autonomous Wi-Fi-based home security system was compared to popular commercial smart security platforms, namely Ring and Google Nest. Four essential aspects were focused on: motion detection, sensing accuracy, alerting latency, and video capture performance. The commercial systems are equipped with cutting-edge features like AI-based human detection, two-way audio, infrared night vision, and cloud analytics. They also offer optional professional monitoring and a variety of proprietary sensors, such as those for doors, windows, and glass breaks. However, these systems operate within closed architectures, which can limit how much you can customize them and how transparent their algorithms are.

The designed / developed system takes a different approach by utilizing an open, modular architecture that activates real-time video capture and notifies the owner instantly via Wi-Fi when PIR-based motion detection is triggered. This setup is designed for easy expansion, allowing the integration of OpenCV-based computer vision modules for optional features like face recognition, object tracking, and anomaly detection. What sets this system apart from commercial options is that it grants full access to both hardware and software layers, making it possible to modify algorithms and optimize the entire system.

Table 5: Performance Comparison Results

Metric	Ring system	Nest system	Designed system
Motion detection Accuracy (%)	96.2	97.5	93.8
Alert latency (ms)	420	380	520
Video Capture delay (ms)	310	290	450
System reliability (%)	98.1	98.7	95.4
False alarm rate (%)	3.5	2.8	5.2

The developed system shows an impressive detection accuracy of over 93,8% and keeps alert latency below 600 ms, making it perfect for real-time monitoring as shown in Table 5. While there are some slightly higher false alarm rates due to the sensitivity of the PIR sensors, we can tackle this issue with computer vision filtering. Overall, the reliability exceeds 95.4%, which really highlights the strong performance of the system.

ii. Cost Performance Analysis

TABLE 6 presents a cost performance analysis which shows that the developed system delivers similar functional capabilities but at a much lower cost.

Table 6: Cost Performance Analysis

System	Initial Hardware Cost (USD)	Subscription fee	Expandability	Vendor
Ring Alarm Kit	199–250	100–200	Limited	High
Google Nest Secure	350–450	120–240	Limited	High
Developed System	120–180	0	High	None

The data shows that our designed system decreases total costs by approximately 40–65% when compared with commercial options, and it also does away with those recurring subscription fees. All things considered, this Wi-Fi home security system provides, at comparable or less cost, the basic capabilities of commercial systems (real-time alarms, video streaming, and mobile app control). It gives up plug-and-play simplicity in favor of customizing possibilities free from subscription costs. This system is particularly fit for tech-savvy consumers or small apartment environments where a totally custom, reasonably priced solution is desired.

V. Conclusion

A thorough design concept is presented for an autonomous home security system based on Wi-Fi ideal for flat use. The system detects and alerts real-time intrusion by combining PIR motion sensors and camera modules with a Raspberry Pi including Arduino and ESP32 modules. By means of MQTT and REST technologies, the mobile app offers remote monitoring, control, and user-centric notifications. Every design decision which includes hardware component, software architecture, UI layout is justified. This system provides competitive performance when compared to current solutions. It offers necessary security features at a reduced hardware cost with great extensibility for future developments. Although the implementation calls for more setup work, it gives the user complete control over privacy and design expansion. Future work intends to improve its low-light camera performance, and maybe add artificial intelligence-based elements like facial recognition. In the end, this idea offers a workable blueprint for self-built smart home security, tying academic IoT research into useful, user-oriented design.

VI. Acknowledgements

The authors wish to thank the Federal Ministry of Education, Nigeria, through the TETFUND Institution Based Research (IBR) Intervention funding for sponsorship of this research work.

References

[1] Vardakis, G., Hatzivasilis, G., Koutsaki, E., & Papadakis, N. (2024). Review of of Smart-Home Security Using Using the Internet of Things. Electronics, 13, 3343. https://doi.org/10.3390/ electronics13163343

[2] Lang, K., Sanford, J., & Murtagh, C. (2025). Assessing CCTV in Preventing and Reducing Property Crime. Justice Evaluation Journal, 8(2), 263–283. https://doi.org/10.1080/24751979.2025.2474706

[3] Magara, T., & Zhou, Y. (2023). Internet of Things (IoT) of Smart Homes: Privacy and Security. Journal of Electrical and Computer Engineering, 2024(1), 7716956. https://doi.org/10.1155/2024/7716956

[4] Nivedha, A., Soundariya, V., Jenifer, C., & Karishma, S. (2018). Internet of Things - Smart Surveillance System using PIR Sensor with Raspberry Pi. International Journal on Future Revolution in Computer Science & Communication Engineering, 4(4), 22–24.

[5] McDowell, Y. (2020). Smart Security Cameras:The Corporatization of the Surveillant Assemblage. The Journal, 5(2).

[6] Prasad, S., P.Mahalakshmi, &A.John Clement Sunder, R. S. (2014). Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor. International Journal of Computer Science and Information Technologies, 5(August), 7107–7109.

[7] Mahyan, F. B., Lit, A. Bin, Satu, N. A., Banyah, M. S. A., &Gumis, M. I. A. (2024). Raspberry Pi-based home security monitoring system. E3S Web of Conferences, 479, 0–7. https://doi.org/10.1051/e3sconf/202447907014

[8] Mohamed, K., & El Shenawy, A. (2023). A Smart IoT-Based Home Automation System for Controlling and Monitoring Home Appliances. International Review of Automatic Control, 16(5), 228–237. https://doi.org/10.15866/ireaco.v16i5.23872

[9] Ajayi, O., Izang, A. A., Osuji, C. F., Umeozo, C. T., & Albert-sogules, T. (2023). Design and Implementation of a WiFi-Enabled Home Automation System. Journal Européen Des Systèmes Automatisés, 56(5), 849–855.

[10] Gaonkar, A., &Negalur, G. (2023). Smart Security Design of Home Automation and Security System Using Raspberry Pi. International Journal for Multidisciplinary Research (IJFMR), 5(6), 1–14.

[11] Silvia Ganesan, Than Yin Ying, Parvenkumar Ravi, C. P. L. (2022). Designing an Autonomous Triggering Control System via Motion Detection for IoT Based Smart Home Surveillance CCTV Camera (pp. 80–88).

[12] Netinant, P., Utsanok, T., &Rukhiran, M. (2024). IoT Development and Assessment of Internet of Things-Driven Smart Home Security and Automation with Voice Commands. 79–99.

[13] Ahmed, M., & Akhtar, M. M. (2021). Smart Home: Application using HTTP and MQTT as Communication Protocols. https://arxiv.org/abs/2112.10339

[14] Ojokoh, P., & Agbolade, O. (2025). A Solar-Powered Multimodal IoT Framework for Real-Time Transformer Theft Detection.

[15] P. Xu, D. Xia, B. Zheng, L. Huang and L. Xie, "A Novel Compensatory Motion Detection Method Using Multiple Signals and Machine Learning," in IEEE Sensors Journal, vol. 22, no. 17, pp. 17162-17172, 1 Sept.1, 2022, doi: 10.1109/JSEN.2022.3190503.

[16] Saleem, M., Ortiz-garcés, I., & Villegas-ch, W. (2025). Autonomous cyber-physical security middleware for IoT : anomaly detection and adaptive response in hybrid environments. December, 1–24. https://doi.org/10.3389/frai.2025.1675132

[17] Bhandari, Sabin & Sharma, Shree Krishna & Wang, Xianbin. (2017). Latency Minimization in Wireless IoT Using Prioritized Channel Access and Data Aggregation. 10.1109/GLOCOM.2017.8255038.