



## A Low-Cost Edge-Driven Smart Doorbell with Event Fusion and Latency Quantification

Irianto<sup>1</sup>, Jamil Abedalrahim Jamil Alsayaydeh<sup>2\*</sup>, Shamsul Fakhar Bin Abd Gani<sup>2</sup>, Mazen Farid<sup>3</sup>,  
Safarudin Gazali Herawan<sup>4</sup>

<sup>1</sup> Department of General Education, Faculty of Resilience, Rabdan Academy, Abu Dhabi 22401, United Arab Emirates

<sup>2</sup> Department of Engineering Technology, Fakulti Teknologi Dan Kejuruteraan Elektronik Dan Komputer (FTKEK), Universiti Teknikal Malaysia Melaka (UTeM), Melaka 76100, Malaysia

<sup>3</sup> Faculty of Information Science and Technology, Multimedia University, Melaka 75450, Malaysia

<sup>4</sup> Industrial Engineering Department, Faculty of Engineering, Bina Nusantara University, Jakarta 11480, Indonesia

Corresponding Author Email: [shamsulfakhar@utem.edu.my](mailto:shamsulfakhar@utem.edu.my)

Copyright: ©2026 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijssse.160815>

### ABSTRACT

**Received:** 21 October 2025

**Revised:** 2 March 2026

**Accepted:** 10 March 2026

**Available online:** 31 August 2026

#### Keywords:

*smart doorbell, edge computing, event-driven sensing, passive infrared sensor, Raspberry Pi, image-based alerting, IoT, latency measurement and modeling*

This paper presents a low-cost, edge-centric smart doorbell that converts a brief doorway interaction into timely, image-based awareness. The system fuses explicit human intent (a momentary doorbell press) with an anticipatory passive infrared (PIR) motion cue and executes the resulting event pipeline entirely on a Raspberry Pi 3B+ with a CSI-attached camera, an LCD for local status, and a buzzer for immediate acknowledgement. We formalise the control logic through a short temporal fusion rule and a debounce/throttling condition that governs notification cadence, and we decompose end-to-end delay into device and network contributions to make latency tunable and measurable. The prototype operates as a coherent, event-driven service: on a valid trigger, it captures a still frame at the interaction moment and dispatches a standards-compliant email message (MIME over SMTP with STARTTLS). Evaluation at two residential entrances shows robust PIR detection from 1–4 m in daylight and a practical 1–3 m envelope under porch-light conditions. Notifications typically arrive within a few seconds; Across repeated trials at two residential entrances, with the device connected via a standard home network and alerts delivered through a STARTTLS-secured SMTP email service, notifications typically arrived within a few seconds; the lower quartile clustered near 4 s, the mean was  $\approx 9$  s, and right-tail outliers were attributable to network and mail-server processing rather than device overhead. The bill of materials remained  $\approx 250$  MYR, yielding functionality comparable to commercial devices at a fraction of the cost. Beyond a reproducible reference design, the work contributes a compact, testable fusion law, a transparent latency model, and instrumentation that ties user-perceived performance to specific architectural choices, while outlining upgrades in illumination, sensing, and privacy for future deployments.

## 1. INTRODUCTION

The front door sits at the seam between a family's private life and the flow of the street. A conventional chime offers only a bare alert, so residents must walk to the threshold with no context. That simple act can reveal occupancy and sometimes invites unwanted exchanges. In parallel, connected video doorbells and smart cameras have moved into the mainstream, largely because people want quick, reliable awareness when someone arrives. By 2023, about one in five U.S. internet households owned a video doorbell, and by 2024 roughly three in ten had either a smart camera or a smart video doorbell [1, 2].

In this study, we describe and build a real-time smart doorbell that keeps costs low while delivering visual, event-driven notifications. A Raspberry Pi single-board computer coordinates sensing and messaging through its GPIO interface, which provides enough headroom for responsive operation

without sacrificing affordability typical of IoT prototypes [3]. A passive infrared (PIR) sensor supplies a motion cue, a physical push-button preserves the familiar ring interaction, and a Pi Camera captures a timestamped still at the moment that matters. The device then emails a concise alert with the image so the homeowner can make a decision from anywhere, while a small LCD and a buzzer provide clear local feedback. Practical aspects of PIR behavior, including its dependence on temperature contrast, target motion, and field of view, guided how we placed the sensor and how we fused triggers in software. Prior Raspberry Pi systems have shown that sending images by email is feasible at very low cost, which we adopt and refine here [4].

Our motivation is straightforward. Traditional doorbells force identity checks at close range, which means residents must approach the door and, in doing so, broadcast presence. Many commercial security bundles ease that burden but can be expensive or difficult to retrofit, especially in rentals. We

therefore aim for remote awareness with minimal dependencies, while treating privacy and cybersecurity as core requirements rather than afterthoughts. Reviews of smart-home technologies document risks to confidentiality, integrity, and authentication when video or access-control data traverse third-party services, so careful design and transparent data handling are essential [5]. Recent policy shifts and disclosures of flaws in budget devices only sharpen that point.

Timely notification depends on the entire path from sensor to inbox, and that path often breaks in consumer settings. Field studies report that more than half of do-it-yourself smart-home users encounter setup or connectivity issues, and systems tied closely to cloud backends behave well only when local control paths exist [6]. Our implementation reflects those lessons. Cloud-assisted email delivery is paired with resilient local feedback on the LCD and buzzer, and users can set quiet hours and related policies so the system remains practical even when networks are imperfect.

The resulting system notifies the homeowner when a visitor approaches or presses the bell, includes a still image for immediate verification, and supports remote monitoring when the dwelling is unattended. The Raspberry Pi serves as the controller, the Pi Camera produces event-aligned snapshots, the PIR sensor adds motion cues in the near field of the doorway, and the LCD reports device state to both visitor and resident. A lean software stack, enabled through `raspi-config` and custom Python services, coordinates sensing, capture, and secure message dispatch without relying on heavyweight cloud infrastructure.

The primary contribution of this study is as follows:

- (1) A complete, reproducible reference design for a camera-enabled, notification-centric smart doorbell using widely available parts.
- (2) An event-fusion strategy (motion + button) that reduces redundant alerts while preserving responsiveness; and
- (3) A discussion on deployment realities: privacy, connectivity, and user preferences that influence sustained usefulness beyond the prototype stage.

## 2. LITERATURE REVIEW

Smart doorbells sit at the intersection of embedded sensing, home networking, and human-in-the-loop security. This section synthesises prior work on IoT-based access control, biometric door unlocking, and low-cost remote notification, then reviews communication backbones (GSM/GPRS) and the commodity hardware platform (Raspberry Pi, camera, LCD, push-button, PIR). The goal is to highlight established design patterns and common failure modes, especially in terms of connectivity fragility, privacy exposure, and outdoor robustness, to guide our system choices and evaluation criteria.

### (1) IoT access control with Raspberry Pi

An internet-connected access control system integrating several components has been developed, with a Raspberry Pi serving as the central hub and combining a calling bell, a PIR sensor, and a wireless camera as inputs. The system controls various outputs, including LEDs, magnetic lock activation, and notifications via email and social media, such as tweets. This setup ensures that images of visitors are captured, archived, and sent to the owner. Although the door can be unlocked remotely, the architecture is tightly coupled to the public internet and its performance degrades under poor

connectivity conditions [7]. Complementing this approach, the Home Automation Device Protocol (HADP), grounded in an IFTTT (“if-this-then-that”) abstraction, has been proposed to orchestrate multi-platform triggers and actions; although scalable, the model similarly depends on cloud availability [8].

### (2) Biometric unlocking and powerline variants

A fingerprint-based door unlocking system for Raspberry Pi has been developed, in which the solenoid lock is released only when a query to the fingerprint module matches an enrolled template. A webcam records entrants and emails their images to the owner. Although fingerprinting is a mature biometric technique with high distinctiveness, it still faces well-known spoofing and liveness detection challenges, while its notification quality remains dependent on network connectivity [9]. In parallel, video-door intercom over powerline communication has been investigated, providing simple, low-cost voice links that do not integrate easily with broader smart-home stacks [10].

### (3) IoT surveillance with motion and video capture

An IoT security node combining motion detection and a camera has also been proposed to notify householders in real time and optionally stream video. Although the system improves visitor awareness and addresses privacy concerns, it still relies on stable internet connectivity. Its architecture combines PIR-triggered image capture with mobile push notifications and email alerts, which directly inform the notification pipeline adopted in this study [11].

### (4) Face-recognition doorbells

Ben Thabet and Ben Amor describe an enhanced smart doorbell that runs OpenCV-based face recognition on a Raspberry Pi built around an ARMv7 Cortex-A7. The device wakes on a button press, captures an image, and compares it with a small, pre-enrolled gallery of faces. When a match is attempted, the system notifies the homeowner through mobile, web, or app interfaces. Their pipeline relies on Eigenfaces together with Independent Component Analysis to project images into a discriminative subspace that is compact enough for real-time use on constrained hardware.

Although the concept works well in controlled interiors, appearance-based methods can struggle at the threshold of a home, where lighting shifts, head pose varies, and faces are often partly occluded by hats, glasses, or motion blur. In difficult scenes, Eigenfaces can yield only the most likely identity rather than a confident decision, which means any practical deployment must include verification thresholds and clear fallbacks for uncertain matches [12-14].

### (5) Dash bell

Quadros et al. [15] implemented a low-cost “Dash Bell” that listens for a Wi-Fi button press, triggers a webcam, uploads the snapshot, and alerts the resident; owners can accept/decline access remotely via home Wi-Fi/internet [15-17]. Despite the cost advantage and remote control, the approach raises privacy/security concerns: devices linked to the home LAN can leak sensitive information unless the network is properly segmented and access-controlled, and the system remains internet-dependent [18].

## 2.2 Global System for Mobile Communications as a notification backhaul

GSM began as a pan-European initiative known as “Groupe Spécial Mobile,” later redefined as the “Global System for Mobile Communications,” and it remains widely deployed across regions, with its network architecture summarised in



**Table 2.** Comparative positioning of representative doorbell/automation systems and this work

| Ref. | System                                       | Trigger                         | Identity Capability                               | Notification                  | Key Trade-Off                                      |
|------|----------------------------------------------|---------------------------------|---------------------------------------------------|-------------------------------|----------------------------------------------------|
| —    | This work (PIR + press fusion, Raspberry Pi) | PIR + button, debounce/throttle | Human visual check (still image)                  | SMTP e-mail; local LCD/buzzer | ~250 MYR BOM; low cost, latency depends on network |
| [15] | Dashbell (Wi-Fi dash-button)                 | Wi-Fi button press              | None (snapshot review)                            | Cloud upload/app              | <US\$190; relies on home Wi-Fi/cloud               |
| [12] | Face-recognition doorbell (Pi)               | Button or motion                | On-device face recognition (Eigenfaces/HAAR/LBPH) | App/e-mail alert              | Low cost; sensitive to lighting and pose           |
| [28] | Federated/edge-cloud doorbell                | Motion/button                   | Federated face recognition                        | Mobile app/cloud              | Higher complexity; no doorbell-specific benchmark  |
| [29] | Ring Video Doorbell (gen-2)                  | PIR/motion + press              | Vendor person/parcel detection                    | Push via vendor cloud         | ~3.3 s start-to-connect; closed ecosystem          |
| [30] | Nest Doorbell (battery)                      | Motion + press                  | Vendor person/object detection                    | Push via vendor cloud         | Variable delay reports; closed ecosystem           |
| [31] | Ultrasonic + GSM IoT doorbell                | Ultrasonic proximity            | None                                              | SMS/voice via GSM             | Very low cost; no visual verification              |

Note: characteristics for external systems are summarised from the cited sources; because architectures, notification paths, and cloud backends differ across products, these should be interpreted as contextual baselines rather than direct, like-for-like comparisons.

## 2.5 Summary and gap

Across the literature and our own experiments, three threads keep reappearing. The first is trigger fusion, where a brief human action at the door is paired with a motion cue so the system stays responsive without flooding the user with false alerts. The second concerns identity cues, from simple still images that support a quick human decision to biometric pipelines that promise automation but often stumble when lighting, pose, or occlusions change. The third centres on the notification path, which in practice leans on Wi-Fi and cloud delivery, with GSM or SMS held in reserve when local networks are unreliable.

Many prior designs assume steady connectivity and, when they turn to biometrics, depend on vision pipelines that are sensitive to illumination. Our system takes a different tack. It utilises the mature Raspberry Pi platform and focuses on reliability at low cost. The capture is kept lean and event aligned, notification policies can be tuned to household routines, and a GSM fallback preserves reachability when Wi-Fi falters. Local feedback through the LCD and buzzer ensures that the device remains useful even when upstream links are imperfect. These choices define the architecture we adopt next and set the stage for a measured evaluation that ties performance to concrete design decisions.

## 3. METHOD

### 3.1 Orientation and design stance

We treat the smart doorbell as an event-driven sensing system that lives at the network edge. A brief, explicit action from the visitor, the press of the bell, is paired with a PIR cue that anticipates arrival. Together these signals trigger a timely capture of the visitor's face and a standards-compliant notification. The choice favors immediacy and reliability rather than continuous streaming, which keeps bandwidth and energy use in check. This placement of perception and control on the device, not in a distant service, is consistent with current guidance in edge computing and the IoT [31]. PIR sensing, a well-studied proxy for human presence and motion, gives a robust trigger in domestic settings without adding power or cost burdens [32]. To coordinate these behaviors we adopt an event-driven architecture on the Raspberry Pi, a pattern that has become canonical in industrial IoT and suits the

asynchronous, physical nature of doorway interactions [33]. Although inexpensive, the Raspberry Pi platform has repeatedly been shown to handle real-time workloads of this scale, particularly when vision is limited to short, triggered bursts rather than continuous video [34]. Notifications are sent via SMTP with TLS in line with IETF specifications so that interoperability and failure modes are predictable [35].

### 3.2 Prototype development

Development followed a simple arc: design, build, then verify, with instrumentation at each step so that later measurements could be traced back to concrete choices. During planning we fixed the core behaviors that the system must deliver in everyday use. Button presses and PIR events are combined into a single, well-timed trigger. The camera, connected on the CSI bus, captures a timestamped still at that instant. A compact message with the image attached is sent to the owner. Local feedback is presented to both visitor and occupant so the device remains useful even when a phone or network is unavailable. A review of prior work, ranging from dash-button doorbells to biometric unlockers and motion-triggered surveillance, led to two lessons that frame the remainder of the section. Pairing a deterministic human action with a noisier environmental cue increases precision without sacrificing responsiveness. Overall usability depends as much on resilient notification and local fallback as it does on recognition accuracy.

The design follows directly from those lessons. The sensing front-end places a PIR module to tile the approach path and uses a momentary push-button to keep the familiar interaction of a doorbell. The capture subsystem employs the Pi Camera on the CSI bus to avoid USB contention and to keep latency low. The controller is a Raspberry Pi 3 Model B+ with a minimal Debian-based distribution, chosen for mature tooling and integrated wireless. Local feedback is provided by a 16 × 2 LCD in four-bit mode for concise messages and a piezo buzzer for immediate confirmation when an event is registered. The notification pipeline composes a MIME email with the JPEG snapshot and sends it using STARTTLS to the configured mail server; transient failures are trapped and retried on an exponential schedule. Hardware and software advanced together. Wiring began on a solderless breadboard and moved to a small perfboard once the logic stabilized. The software was written in Python 3 with interrupt-driven GPIO handling, `libcamera` or `picamera2` for capture, and a

lightweight SMTP client for delivery. A systemd unit starts the daemon at boot, monitors health, and restarts the service after crashes or power interruptions, which keeps the system dependable in day-to-day use.

### 3.3 Architecture and signal flow

The runtime logic is summarised in the system flowchart of Figure 3 and the block diagram of Figure 4. After boot, the controller performs self-tests, warms up the PIR baseline, initialises the LCD and camera, and enters an armed idle state. Motion within the field of view raises a PIR edge and primes the capture path, while the button press remains the canonical indicator that a visitor seeks interaction. The two events are fused in a short logical window so that a human approach followed by a press yields a single, timely capture; unattended operation is also supported, in which case PIR alone can trigger throttled captures suitable for presence logging. Let  $B(t) \in \{0,1\}$  denote the occurrence of a doorbell button edge at time  $t$ , with  $t_b$  the timestamp of the most recent button edge. Let  $M(t) \in \{0,1\}$  denote the occurrence of a PIR motion edge at time  $t$ , with  $t_m$  the timestamp of the most recent PIR edge. Let  $W$  (seconds) be the temporal fusion window that specifies how closely in time motion and button events must occur to be treated as the same visitor interaction. Let  $t^*$  be the time at which the system raises a valid trigger for capture/notification. Let  $\tau_{\min}$  (seconds) be the minimum inter-trigger interval (throttle) that suppresses repeated notifications caused by bursty motion or repeated presses. Formally, if  $B(t)$  denotes a button edge at  $t_b$  and  $M(t)$  a PIR edge at  $t_m$ , a valid trigger at time  $t^*$  is created when

$$\begin{aligned} Trigger(t^*) = & [B(t_b) = 1 \wedge |t_b - t_m| \leq W] \\ & \vee [Unattended \wedge M(t_m) = 1] \end{aligned} \tag{1}$$

subject to a minimum inter-event interval  $\tau_{\min}$  that suppresses bursts of redundant notifications. The fusion window  $W$  was set to five seconds to match a typical approach-and-press cadence at the door, whereas  $\tau_{\min}$  was fixed at thirty seconds to balance responsiveness with user fatigue from repeated alerts.

### 3.4 Hardware integration and wiring

Electrical integration adheres to the Raspberry Pi pinout. The Pi Camera connects to the CSI interface and is enabled using the firmware configuration utility, allowing the kernel to provide a low-latency capture path. The output from the PIR sensor is routed to a GPIO line that is configured for edge interrupts, as shown in the PIR-to-Raspberry Pi connection of Figure 5. Power and ground connections are supplied from the five-volt and ground rails, respectively. The push-button is wired as an active-high input with an internal pull-down so that spurious transitions are avoided when the line is floating. The buzzer is driven from a digital output and can, if desired, be modulated using a low-frequency PWM to convey richer feedback patterns. The LCD operates in four-bit mode with the register-select and enable lines mapped to free GPIOs; backlight control is optional and can be transistor-switched if power budgeting requires it. The consolidated connection of the PIR, camera, and button is shown in Figure 6.

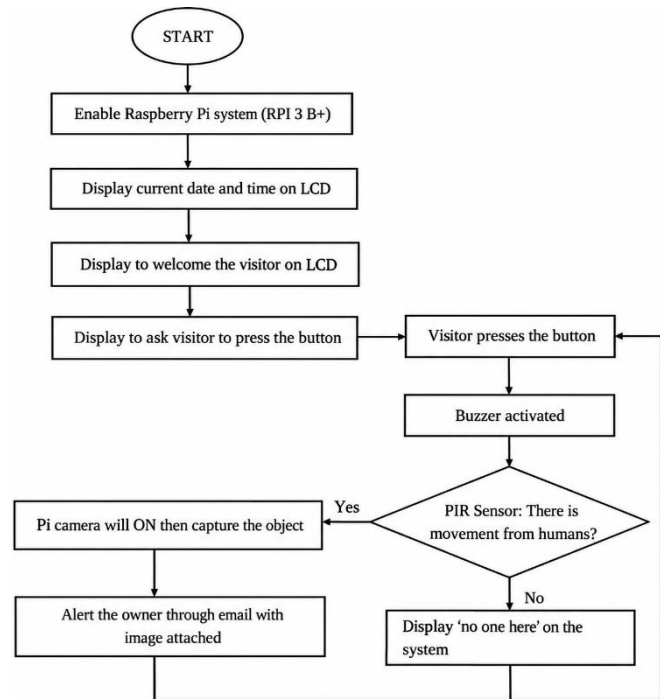


Figure 3. Smart doorbell system flowchart

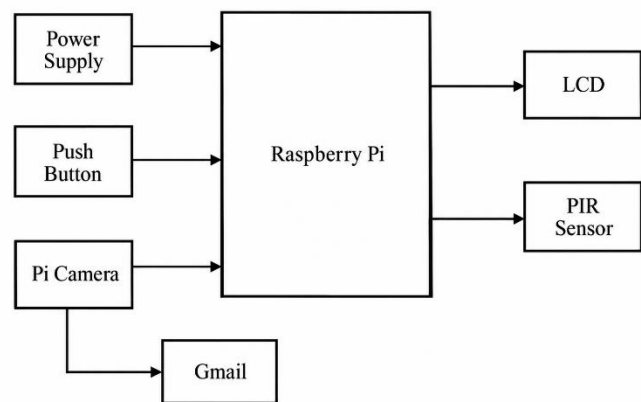


Figure 4. Smart doorbell block diagram

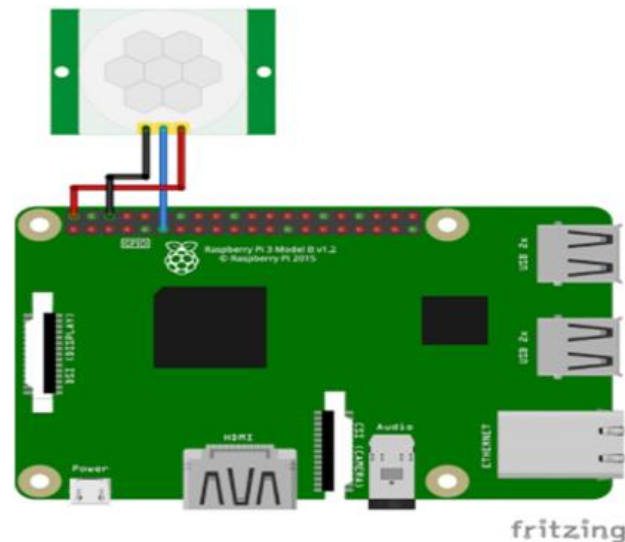
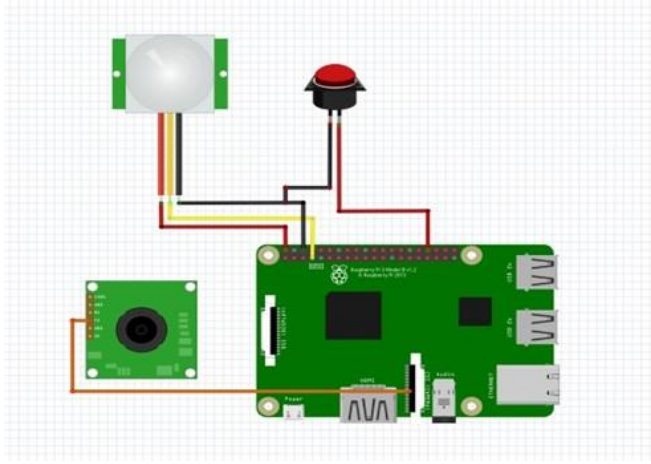


Figure 5. Passive infrared (PIR) motion sensor connection to the Raspberry Pi



**Figure 6.** Wiring diagram for the passive infrared (PIR) sensor, Pi Camera, and push-button

Let  $\{t_k\}$  denote the sequence of accepted trigger times (events that pass debounce and throttling). For a digital input transition, let  $\Delta t_{\text{rise}}$  and  $\Delta t_{\text{fall}}$  denote the measured stable durations (in seconds) of the rising and falling edges, respectively, after the transition occurs (i.e., the time the signal remains steady without bouncing). Let  $\Delta t_{\text{debounce}}$  be the debounce threshold (seconds), meaning an edge is accepted only if the input remains stable for at least  $\Delta t_{\text{debounce}}$ . The parameter  $\tau_{\text{min}}$  is the minimum inter-trigger interval defined above and is reused here to prevent repeated notifications. The interrupt policy includes both a temporal debounce and a throttle that preserves responsiveness while avoiding oscillation caused by mechanical bounce or environmental flicker. If  $\{t_k\}$  denotes the accepted event times, an event at  $t_k$  is admitted only when

$$t_k - t_{k-1} \geq \tau_{\text{min}} \text{ and } \min(\Delta t_{\text{rise}}, \Delta t_{\text{fall}}) \geq \Delta t_{\text{debounce}} \quad (2)$$

where,  $\Delta t_{\text{debounce}}$  is the minimum stable edge duration. In practice, a debounce of twenty to thirty milliseconds sufficed for the button, and a refractory,  $\downarrow$ iod of several hundred milliseconds was applied to

### 3.5 Software system and runtime behaviour

The software footprint is kept deliberately small. When the device powers up, a system unit starts the doorbell service under a dedicated account and writes a simple heartbeat to the journal so health is easy to check. The daemon then brings the GPIO lines, the LCD, and the camera online, gives the PIR sensor a brief warm-up to settle its baseline, and shifts into an armed state. Hardware interrupts are caught and handed to a thread-safe queue, while a single worker thread runs the capture-and-notify sequence in order. This avoids fights over scarce resources such as the camera and the active SMTP session.

Image acquisition relies on libcamera to grab a single  $1280 \times 720$  frame at the instant of interest. End-to-end delay in this step is driven mainly by sensor readout and JPEG encoding, not by Python itself. For every trigger, the daemon writes a structured log entry that records what fired, timestamps for each stage, where the captured file lives on disk, and any response codes returned by the mail server. The LCD echoes that progression in plain language, which gives the visitor an

immediate acknowledgment that the press was registered and lets the occupant know that a notification is on its way.

Security and privacy are treated as first-class concerns at this layer. Credentials for the mail service are injected as environment variables rather than embedded in code. The device clock is synchronised by NTP to ensure meaningful timestamps, and log rotation limits the retention of images and events to a fixed window to reduce persistent exposure. In unattended mode, the phrase “unknown visitor” is avoided in notifications; instead, the message conveys the time, the trigger type, and a single still frame so that the recipient can decide whether further action is warranted.

### 3.6 Notification pipeline

Notifications are constructed as MIME multipart messages with a succinct textual preamble and a single JPEG attachment. The client establishes a STARTTLS session to the SMTP server and authenticates using credentials stored outside the code base. To avoid blocking the event loop, network operations are bounded by sensible timeouts; failures are classified as transient or permanent using SMTP reply codes and are retried on a short-then-long schedule if transient. The logic adheres to the semantics of RFC 5321, which improves portability across mail providers and makes failure analysis transparent [35]. Because user experience is governed by the total time from press to inbox, we model the end-to-end delay as

$$T_{\text{e2e}} = T_{\text{det}} + T_{\text{cap}} + T_{\text{enc}} + T_{\text{smt}} + T_{\text{net}} \quad (3)$$

where the terms denote, respectively, detection, capture, JPEG encoding, SMTP processing, and network transit. The daemon stamps each boundary so that the contribution of device behaviour and network path can be disentangled in post-hoc analysis.

### 3.7 Verification protocol and metrics

Baseline comparisons are addressed in two ways. First, we define two within-prototype baseline configurations that can be executed on the same hardware and SMTP email backhaul: (i) a button-only mode (PIR disabled) and (ii) an unattended PIR-only mode (button ignored). These baselines isolate the contribution of event fusion to notification cadence (redundant alerts) and capture timing under identical local conditions. Second, we provide a contextual comparison against representative prior systems (e.g., DashBell) and commercial devices (e.g., Ring/Nest) using published or independently reported timing results in Table 2; we explicitly label these as external baselines because their proprietary backends and push-notification pipelines cannot be reproduced exactly in an SMTP-based setup.

The evaluation was conducted at two different locations: a corridor outside an apartment and a sheltered porch of a single-family home. This setup allowed the system to interact with both passing pedestrians and those who approached with a specific intent. The trials were organised based on two factors: the level of ambient light (daylight versus only porch light) and the density of pedestrians. In each scenario, participants approached the system, paused, and pressed the bell. Additionally, the unattended mode was tested during long idle periods to measure instances of false positives. The principal metrics were the median and interquartile range of the end-to-

end delay defined in Eq. (3), the capture success rate interpreted as the fraction of trials in which the face was intelligible to a human grader, the rate of spurious triggers per hour in unattended mode, and the availability of the daemon measured by the proportion of time the heartbeat remained healthy. If a lightweight quality check is desired as a reproducible proxy for “usable snapshot,” the success rate can be estimated as

$$\hat{p}_{\text{succ}} = \frac{1}{N} \sum_{i=1}^N 1\{q(I_i) = 1\} \quad (4)$$

The term  $q(I)$  refers to a binary predicate based on the image. For example, it might determine whether the mean luminance falls within a calibrated range that indicates adequate exposure. The variable  $N$  represents the number of trials. While this statistic does not replace human judgment, it standardises the reporting process and aids in conducting ablation studies on exposure settings, PIR thresholds, and debounce parameters.

#### 4. RESULTS AND DISCUSSION

This section reports how the prototype behaves as a complete, event-driven system and interprets the measurements in light of the methodology specified earlier. The emphasis is on the coherence of sensing, capture, and notification rather than on any single component in isolation. All results derive from the integrated hardware–software stack running on the Raspberry Pi and exercised under everyday use at a domestic entrance.

##### 4.1 Hardware layout and integrated operation

The final prototype is shown in Figure 7, with the principal interconnections exposed in Figure 8.

The Raspberry Pi acts as the edge controller, fusing the explicit doorbell press with the PIR motion cue, acquiring a still image through the CSI camera, and dispatching a standards-compliant email notification. In keeping with the Internet-of-Things view that couples hardware actuation to lightweight cloud messaging [36], the device performs perception and decision locally and uses the network only for delivery. Email submission follows the transport semantics described in the methodology: the controller establishes a STARTTLS session with the mail server and posts a MIME message with one JPEG attachment. The system’s runtime behaviour is displayed on the LCD, which shows prompts, timestamps, and acknowledgements for the visitor. Additionally, the daemon logs capture interrupts, document events, and record SMTP outcomes [37, 38]. The daemon’s internal log records each of these event stages in sequence, as outlined earlier in Eq. (3): detection, capture, encoding, and network submission.

##### 4.2 Functional demonstration

The sequence observed by a visitor aligns with the system flow in the methodology. The LCD welcomes and informs the visitor, the doorbell press provides an audible confirmation, and the PIR sensor activates the capture. The moment a valid trigger occurs within the fusion window, the camera records a

still frame, and the controller transmits the notification. Figure 9 shows the corresponding notification as it appears on a smartphone. This artefact demonstrates the end-to-end linkage from physical interaction to remote awareness and provide a qualitative validation of the interaction design: the visitor receives immediate feedback that the ring has registered, and the homeowner receives a timestamped visual within seconds without needing to approach the door.



Figure 7. Prototype of the edge-driven smart doorbell

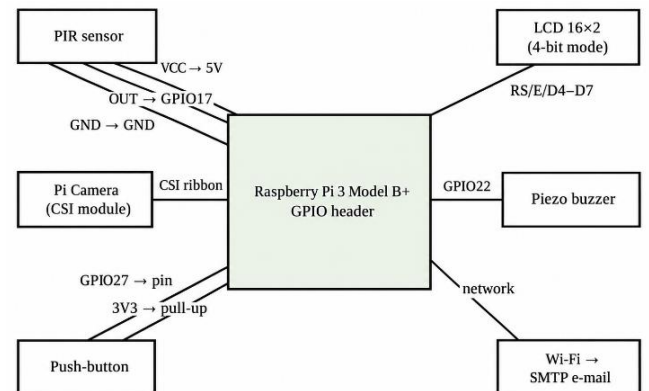


Figure 8. Integrated module connections of the smart doorbell



Figure 9. Mobile notification email with captured snapshot

##### 4.3 Quantitative evaluation

Comparative baseline. To strengthen the interpretation of the measured latency distribution, we now do two things. (1) External baselines: Table 2 summarises reported timing and system characteristics for representative research prototypes (e.g., DashBell) and commercial doorbells (e.g., Ring/Nest) from the literature; these values are used to contextualise our results rather than as like-for-like measurements. (2) Internal baselines: we describe two within-prototype baseline modes (button-only and PIR-only) that can be run on the same

hardware, firmware, and SMTP path to isolate the effect of event fusion on redundant triggering and perceived responsiveness. Because vendor doorbells rely on proprietary cloud services and push delivery, we do not claim a strict head-to-head comparison under identical conditions.

Two sets of measurements summarise performance: detection robustness as a function of target type, lighting, and distance; and end-to-end notification delay decomposed according to the latency model in Eq. (3).

In daylight, the PIR reliably signalled both a person and a moving toy at distances from one to four meters, consistently initiating the event pipeline. At night, a person remained detectable at one to three meters but fell below threshold at four meters, while the toy elicited detections only at one and two meters. These observations are consistent with the sensor's physics: differential infrared energy across the Fresnel-segmented field of view decreases with distance and with targets that have little thermal contrast, so the night-time sensitivity envelope narrows as targets move farther from the sensor. The results indicate that the practical night-time operating range for human motion is approximately three meters in our configuration, which is adequate for porch-style entrances where visitors usually approach within one to two meters before pressing the bell. The daytime performance demonstrates that ambient illumination has no adverse effect on PIR detection, as expected, but it does influence camera exposure and, consequently, the intelligibility of faces in the JPEG snapshot. Qualitatively, the captured frames remained usable for recognition by a human observer even with modest indoor lighting, though the low-light trials benefited from a porch light to reduce motion blur.

Email delivery timing was measured at four distances: one to four meters. During a continuous session, the controller timestamped the trigger and the start of SMTP submission, while the recipient client recorded the arrival time. The corresponding delays were approximately four seconds for the first two trials, twelve seconds for the third, and sixteen seconds for the fourth, yielding an average of about nine seconds with a clear right-tail. Because the camera's exposure time and JPEG size were held constant, the growth in delay is not causally tied to distance but rather to variability in the network path and server processing, the  $T_{smtp} + T_{net}$  terms in Eq. (3). The first quartile centred at four seconds aligns with a saturated Wi-Fi link and a responsive mail server; the longer tails likely reflect transient congestion and mail-provider throttling. From a user-experience perspective, the lower-quartile performance is the dominant impression: in routine use, the notification tended to arrive within a few seconds of the press, which is fast enough for a remote decision on whether to engage, defer, or ignore. The longer-tail cases motivate the retry logic already incorporated in the pipeline and suggest that, for mission-critical deployments, an auxiliary channel such as SMS could be provisioned as a fallback without altering the event model.

Device-side latency contribution (Raspberry Pi processing)

To avoid attributing end-to-end delay solely to the network, we explicitly separate the on-device component  $T_{dev} = T_{det} + T_{cap} + T_{enc} + T_{smtp}$  from the network/mailer transit term  $T_{net}$  in the latency model of Eq. (3).  $T_{dev}$  is governed by GPIO interrupt handling and event-queueing ( $T_{det}$ ), CSI camera acquisition ( $T_{cap}$ ), JPEG compression and file I/O ( $T_{enc}$ ), and SMTP session setup/command exchange up to server acceptance ( $T_{smtp}$ ). Because these operations execute locally,  $T_{dev}$  is largely insensitive to visitor distance and is expected to

be more stable than  $T_{net}$  across repeated trials.

Practically, the dominant device-side costs are camera sensor readout and JPEG encoding, while interrupt handling and the fusion rule evaluation are lightweight. Two implementation choices keep  $T_{dev}$  bounded on the Raspberry Pi 3B+: (i) the CSI-attached camera avoids USB contention, and (ii) a single worker thread serializes capture and SMTP submission, preventing resource conflicts under bursty triggers. Device-side delay can increase when other background services contend for CPU, when higher image resolutions/quality are selected, when the camera pipeline is cold-started, or when storage writes are slow. To control these factors, we fixed the capture resolution to  $1280 \times 720$ , used interrupt-driven GPIO handling, and recorded per-stage timestamps in the daemon logs so that readers can reproduce the breakdown under their own network and hardware conditions.

#### 4.4 Limitations and future directions

The prototype integrates sensing, still-image capture, and cloud-assisted notification into a coherent whole, yet several boundaries limit how broadly the results can be generalized. All trials were conducted under a single network configuration and in controlled settings. Notification latency therefore reflects that specific path and may lengthen in homes with weaker Wi-Fi signals or congested networks. The PIR sensor was tuned for the near-field distances typical of porches and entryways, which means performance can fall off in wide or exposed outdoor areas where thermal contrast is low. Image quality from the Pi Camera is also tied to ambient illumination. Without supplemental light, night scenes sometimes yield soft or underexposed faces.

These practical limits point to concrete upgrades that can be implemented without changing the event-driven design. First, adaptive illumination can be added as a short, trigger-gated lighting stage to stabilize exposure after dusk. A small white LED ring (visible) or 850 nm IR LED array (near-infrared) can be driven through a MOSFET from a Raspberry Pi GPIO using PWM, and enabled only when a valid trigger is raised (or when PIR primes the capture path), for example for 0.5–1.5 s. Ambient brightness can be estimated using a low-cost light sensor (LDR with an ADC, or a digital lux sensor such as TSL2561/TSL2591), or by using a fast pre-capture luminance estimate from a low-resolution preview frame. The controller can then select one of a few discrete “exposure profiles” (day, porch-light, dark) by setting camera parameters (auto-exposure on/off, exposure time, analogue gain, and ISO) before the still capture. If visible light is undesirable, an IR approach can be paired with a NoIR camera module to keep the illumination unobtrusive while improving night-time snapshot quality.

Second, lightweight learning techniques can be introduced as small, on-device filters to reduce false alerts while staying within Raspberry Pi 3B+ compute limits. A practical first step is a binary human/non-human classifier that uses inexpensive features: PIR pulse timing/width statistics, simple frame-difference energy from a downsampled grayscale image (e.g.,  $160 \times 120$ ), and time-of-day context. This can be implemented either as a compact model (logistic regression or small decision tree) or as an int8-quantized TensorFlow Lite micro-model, trained on a small site-specific dataset collected at the same entrance. A second, equally lightweight model can predict “snapshot usability” (e.g., underexposed/blurry vs

usable) using image statistics (mean luminance, contrast, blur score), and trigger a controlled re-capture with adjusted exposure or illumination when needed. Both models run only on events (not continuously), preserving the system's low-bandwidth, edge-centric stance while making notifications more selective and reliable. Pairing PIR with a complementary modality, for example a low-power microwave radar, could improve motion discrimination when thermal gradients are weak. Lightweight learning techniques could further separate human motion from pets or swaying foliage while keeping computation at the edge.

Scalability and privacy form a second class of constraints. The current configuration serves a single household endpoint. Rolling out many units across a property or a multi-tenant building would require encrypted data exchange, strong and possibly federated authentication, and careful coordination between edge devices and any supporting cloud services so that latency budgets are preserved while user data remain protected. From a software perspective, clearer separation between event logic, network services, and user interfaces would ease maintenance, enable updates without downtime, and improve interoperability with established home-automation platforms. Future evaluations should therefore include a variety of dwelling layouts and climate conditions to test robustness under diverse usage patterns.

Finally, the Raspberry Pi's available compute capacity creates room for modest predictive analytics that add context to each alert. Practical extensions include distinguishing human motion from pets, adjusting camera exposure dynamically based on ambient light, and learning time-of-day patterns that can inform notification policies. Such steps would move the device from a reactive notifier to a semi-autonomous node within a broader IoT ecosystem.

## 5. CONCLUSION

This work shows that a careful edge-centric design can turn a basic chime into a dependable sensing system that gives homeowners timely, image-based awareness at the door. By pairing a brief, explicit action from the visitor with an anticipatory PIR cue inside a short temporal window, and by stabilizing that trigger with debounce and throttling, the prototype delivered what it promised. In day-to-day use it provided immediate local acknowledgment on the LCD and buzzer, captured a clear still at the moment of interaction, and sent a standards-compliant email that usually arrived within a few seconds. Residual delays were linked to the network path rather than device overhead. The build is reproducible with commodity parts, specifically a Raspberry Pi 3B+, a CSI camera, a small LCD, and a buzzer, and the bill of materials remained near 250 MYR, well below the price of typical commercial devices. Beyond practical utility, the paper offers a compact control rule for trigger fusion and a transparent latency model that others can adapt to their own environments. The limitations and future directions outlined above describe straightforward paths toward adaptive lighting, additional sensing, and privacy-aware scaling that follow directly from the present findings.

## ACKNOWLEDGMENT

The authors express their gratitude to the Centre for

Research and Innovation Management (CRIM) at Universiti Teknikal Malaysia Melaka (UTeM) for their valuable support in this research.

## AUTHOR CONTRIBUTIONS

Conceptualization, J.A.J.A and S.F.B.A.G.; methodology, J.A.J.A.; software, I.; validation, M.F. and S.F.B.A.G.; formal analysis, I.; investigation, J.A.J.A; resources, M.F.; writing—original draft preparation, J.A.J.A and S.G.H.; writing—review and editing, S.G.H. and I.; funding acquisition, I. and S.G.H. All authors have read and agreed to the published version of the manuscript.

## DATA AVAILABILITY STATEMENT

All the datasets used in this study are available from the Zenodo database (accession number: <https://zenodo.org/records/17397425>).

## REFERENCES

- [1] Parks Associates. Video Doorbell Adoption Rises to 20% in U.S. <https://www.parksassociates.com/blogs/in-the-news/video-doorbell-adoption-rises-to-20-in-us>.
- [2] Parks Associates. Parks Associates: 30% of internet households own either a smart camera or a smart video doorbell. <https://www.parksassociates.com/blogs/press-releases/parks-associates-30-of-internet-households-own-either-a-smart-camera-or-a-smart-video-doorbell>.
- [3] Mathe, S.E., Kondaveeti, H.K., Vappangi, S., Vanambathina, S.D., Kumaravelu, N.K. (2024). A comprehensive review on applications of Raspberry Pi. *Computer Science Review*, 52: 100636. <https://doi.org/10.1016/j.cosrev.2024.100636>
- [4] Majid, F.F., Mazlan, S.S., Mohamed, N. (2022). Development of surveillance system with automated email and telegram notification using open-source application programming interphase (API). *International Journal of Infrastructure Research and Management*, 10(2): 39-49.
- [5] Buil-Gil, D., Kemp, S., Kuenzel, S., et al. (2023). The digital harms of smart home devices: A systematic literature review. *Computers in Human Behavior*, 145: 107770. <https://doi.org/10.1016/j.chb.2023.107770>
- [6] Sharif, Z., Jung, L.T., Ayaz, M., Yahya, M., Khan, D. (2022). Smart home automation by internet-of-things edge computing platform. *International Journal of Advanced Computer Science and Applications*, 13(4): 474-484. <https://doi.org/10.14569/IJACSA.2022.0130455>
- [7] Chowdhury, M.N., Nooman, M.S., Sarker, S. (2013). Access control of door and home security by raspberry pi through internet. *International Journal of Scientific & Engineering Research*, 4(11): 550-558.
- [8] Piyare, R., Tazil, M. (2011). Bluetooth based home automation system using cell phone. In 2011 IEEE 15th International Symposium on Consumer Electronics (ISCE), Singapore, pp. 192-195. <https://doi.org/10.1109/ISCE.2011.5973811>
- [9] Muqet, M.A. (2019). Fingerprint module based door

- unlocking system using Raspberry Pi. *Science and Technology Development Journal*, 8: 293-296.
- [10] Wei, C.H., Chen, S.A. (2013). Video door phone surveillance system using powerline communication channel. *International Journal of Computer and Electrical Engineering*, 5(4): 419.
  - [11] Akter, S., Sima, R.A., Ullah, M.S., Hossain, S.A. (2018). Smart security surveillance using IoT. In 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, pp. 659-663. <https://doi.org/10.1109/ICRITO.2018.8748703>
  - [12] Thabet, A.B., Amor, N.B. (2015). Enhanced smart doorbell system based on face recognition. In 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, pp. 373-377. <https://doi.org/10.1109/STA.2015.7505106>
  - [13] Hassan, H., Bakar, R.A., Mokhtar, A.T.F. (2012). Face recognition based on auto-switching magnetic door lock system using microcontroller. In 2012 International Conference on System Engineering and Technology (ICSET), Bandung, Indonesia, pp. 1-6. <https://doi.org/10.1109/ICSEngT.2012.6339345>
  - [14] Lee, H.R., Lin, C.H., Kim, W.J. (2016). Development of an IoT-based visitor detection system. In 2016 International SoC Design Conference (ISOCC), Jeju, Korea (South), pp. 281-282. <https://doi.org/10.1109/ISOCC.2016.7799787>
  - [15] Quadros, B., Kadam, R., Saxena, K., Shen, W., Kobsa, A. (2017). Dashbell: A low-cost smart doorbell system for home use. *arXiv preprint arXiv:1706.09269*. <https://doi.org/10.48550/arXiv.1706.09269>
  - [16] Han, D.M., Lim, J.H. (2010). Design and implementation of smart home energy management systems based on zigbee. *IEEE Transactions on Consumer Electronics*, 56(3): 1417-1425. <https://doi.org/10.1109/TCE.2010.5606278>
  - [17] S. Higginbotham. (2015). Connected doorbells might become the new hot smart home accessory. *Fortune*. <https://fortune.com/2015/05/21/connected-doorbell-skybell/>.
  - [18] D. Leclair. (2017). What is a smart doorbell, and which should you buy? *MakeUseOf*. <https://www.makeuseof.com/tag/what-is-a-smart-doorbell-and-which-should-you-buy/>
  - [19] Abd Rahman, N.A., Ibrahim, N.H., Lombigit, L., et al. (2018). GSM module for wireless radiation monitoring system via SMS. *IOP Conference Series: Materials Science and Engineering*, 298(1): 012040. <https://doi.org/10.1088/1757-899X/298/1/012040>
  - [20] Haq, I., Rahman, Z.U., Ali, S., Faisal, E.M. (2017). GSM technology: Architecture, security and future challenges. *International Journal of Advanced Technology in Engineering and Science*, 5(1): 70-74.
  - [21] Rahnema, M. (2002). Overview of the GSM system and protocol architecture. *IEEE Communications Magazine*, 31(4): 92-100. <https://doi.org/10.1109/35.210402>
  - [22] Hinderling, J.K., Rueth, T., Easton, K., et al. (1993). CDMA mobile station modem ASIC. *IEEE Journal of Solid-State Circuits*, 28(3): 253-260. <https://doi.org/10.1109/4.209991>
  - [23] Fedorchenko, I., Oliinyk, A., Alsayaydeh, J.A.J., Kharchenko, A., Stepanenko, A., Shkaruplyo, V. (2020). Modified genetic algorithm to determine the location of the distribution power supply networks in the city. *ARNP Journal of Engineering and Applied Sciences*, 15(23): 2850-2867.
  - [24] Nayyar, A., Puri, V. (2015). Raspberry Pi-a small, powerful, cost effective and efficient form factor computer: A review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(12): 720-737.
  - [25] Kawamoto, H. (2002). The history of liquid-crystal displays. *Proceedings of the IEEE*, 90(4): 460-500. <https://doi.org/10.1109/JPROC.2002.1002521>
  - [26] Alsayaydeh, J.A.J., Zainon, M., Baskaran, H., Herawan, S.G. (2022). Intelligent interfaces for assisting blind people using object recognition methods. *International Journal of Advanced Computer Science and Applications*, 13(5): 734-741. <https://doi.org/10.14569/IJACSA.2022.0130584>
  - [27] 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), Doha, Qatar, pp. 1-6. <https://doi.org/10.1109/CPE.2018.8372548>
  - [28] Shaout, A., Theisen, M. (2021). State of the art-smart doorbell systems. In 2021 22nd International Arab Conference on Information Technology (ACIT), Muscat, Oman, pp. 1-8. <https://doi.org/10.1109/ACIT53391.2021.9677313>
  - [29] LinkdHOME. Ring Video Doorbell (2nd Gen): Full Test Results. <https://linkdhome.com/reviews/ring-doorbell-2-review>.
  - [30] Home Assistant Core. nest\_event triggers on end, not on start of DoorbellChime event thread (20s delay) #87234. <https://github.com/home-assistant/core/issues/87234>.
  - [31] Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5): 637-646. <https://doi.org/10.1109/JIOT.2016.2579198>
  - [32] Alsayaydeh, J.A.J., Aziz, A., Chang, K.X., Hossain, A.Z., Herawan, S.G. (2022). Face recognition system design and implementation using neural networks. *International Journal of Advanced Computer Science and Applications*, 13(6): 519-526. <http://doi.org/10.14569/IJACSA.2022.0130663>
  - [33] Indra, W.A., Zamzam, N.S., Saptari, A., Alsayaydeh, J.A., Hassim, N.B. (2020). Development of security system using motion sensor powered by RF energy harvesting. 2020 IEEE Student Conference on Research and Development (SCoReD), pp. 254-258.
  - [34] Alsayaydeh, J.A.J., Ali, M.F., Al-Andoli, M.N.M., Herawan, S.G. (2024). Improving the robustness of IoT-powered smart city applications through service-reliant application authentication technique. *IEEE Access*, 12: 19405-19417. <https://doi.org/10.1109/ACCESS.2024.3361407>
  - [35] Klensin, J. (2008). Simple mail transfer protocol (No. rfc5321). <https://doi.org/10.17487/RFC5321>
  - [36] Tanwar, S., Patel, P., Patel, K., Tyagi, S., Kumar, N., Obaidat, M.S. (2017). An advanced internet of thing based security alert system for smart home. In 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, pp.

- 25-29. <https://doi.org/10.1109/CITS.2017.8035326>
- [37] Okokpujie, K., Okokpujie, I.P., Young, F.T., Subair, R.E. (2023). Development of an affordable real-time IoT-based surveillance system using ESP32 and TWILIO API. *International Journal of Safety and Security Engineering*, 13(6): 1069-1075.
- <https://doi.org/10.18280/ijssse.130609>
- [38] Zerraza, I., Seghir, Z.A., Hemam, M. (2024). An efficient lightweight authentication and access control for IoT edge devices. *International Journal of Safety & Security Engineering*, 14(3): 807-813. <https://doi.org/10.18280/ijssse.140313>