International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

Smart Home – "Aashraya" IoT Automation System

Mehul Vani

Independent Researcher, USA

mehul1.vani@gmail.com

Divya Dadlani

Independent Researcher, India

divya.dadlani08@gmail.com

ABSTRACT: Smart homes increasingly integrate connected sensors, automated control, and intelligent security functions, but fragmented subsystem management can limit the effectiveness of household resource and safety automation. This paper presents Aashraya, an integrated IoT-based smart-home framework that combines gas-cylinder monitoring and automated refill support, overhead water-tank control, and CNN-based intrusion detection within a unified architecture. The proposed system uses sensor nodes, MQTT-based communication, edge processing on a Raspberry Pi, database services, and a mobile application for monitoring and alert delivery. Gas and water resource-management behaviour was evaluated using the reported household usage scenarios, while the security module was evaluated using a 700-image dataset containing authorized-resident and unknown-visitor images.

The reported results show that average gas-outage duration decreased from 2.3 to 0.2 h/month, corresponding to a 91.3% reduction in outage duration, while the automated gas-management process provided a multi-day replacement buffer. For water management, pump on/off errors decreased from 40% under the baseline condition to 0% with the proposed system, and average pump energy consumption decreased from 480 to 384 Wh/day, corresponding to a 20.0% reduction. On the reported 140-image test set, the intrusion classifier achieved 95.0% accuracy, 90.2% precision, 92.5% recall, and a 91.4% F1 score. MQTT sensor-to-database communication exhibited a reported round-trip latency of 50 ± 10 ms under the evaluated multi-sensor configuration.

The findings demonstrate the feasibility of integrating household resource automation, intrusion monitoring, and low-latency IoT communication within a unified smart-home platform. However, the evaluation is limited by the controlled test environment, partly simulated resource-usage scenarios, and the relatively modest security dataset. Further validation using longer-duration real-world deployments, larger and more diverse datasets, and broader sensor configurations is required to establish generalizability and large-scale deployment performance.

KEYWORDS: Smart Home; Internet of Things; Automation; gas management; water management; intrusion detection; machine learning.

I. INTRODUCTION

Smart homes combine connected sensors, actuators, software services, and user interfaces to improve household automation, security, and resource efficiency [1]. Existing solutions often address isolated functions—such as gas monitoring, water-level control, or surveillance—without coordinating them within one platform. This fragmentation can limit interoperability and usability, particularly where connectivity, technical support, energy, or water resources are constrained.

The proposed system, Aashraya, addresses these limitations through an IoT-based framework that integrates gas-cylinder monitoring and automated reordering, water-tank level control, and physical-intrusion detection. It combines threshold-based control, consumption forecasting, and machine-learning classification with ESP32 controllers, a Raspberry Pi edge gateway, and an Android application for real-time monitoring, control, and alerts. The system model includes resource-state equations, a probabilistic intrusion classifier, and performance objectives for resource management. The reported evaluation uses a hardware-software prototype, partly simulated household usage profiles,

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

and manual or static-threshold baselines. Aashraya is intended to reduce user effort, improve safety, and conserve household resources under the evaluated conditions [2].

II. LITERATURE REVIEW

Smart-home automation combines connected sensors, actuators, mobile interfaces, edge computing, and local or cloud services. General IoT research identifies lightweight communication, interoperability, security, and application integration as central design concerns [9], while edge computing moves latency-sensitive processing closer to devices [10]. Taiwo and Ezugwu [1] and Yar et al. [2] demonstrated integrated Android- and Raspberry Pi-centered platforms for automation, monitoring, safety, and security. Together, these studies show the value of unified platforms while highlighting privacy and resource constraints.

Gas management: LPG-focused IoT research addresses condition monitoring, leak detection, emergency response, and refill notification. Hasan and Ahammed [8] described remote LPG monitoring with automated emergency controls. Mariselvam and Siva Dharshini [11] demonstrated gas-level detection and automatic refill notification using a NodeMCU-connected sensing unit. Aashraya additionally uses load-cell measurements and consumption trends to estimate when reordering should begin.

Water management: Rojek et al. [7] described distributed sensing, Wi-Fi/MQTT communication, and self-refilling tank simulation. Gautam et al. [12] presented ultrasonic sensing and automatic pump control between lower and upper tanks. Jan et al. [13] reviewed IoT-based level monitoring, leakage detection, automatic refilling, and motor control. These studies support connected sensing and actuation while motivating reliable hysteresis-based operations under variable demand.

Security and intrusion: Smart-home security spans network intrusion, physical burglar detection, and camera-based identity recognition. Reviews describe IoT attack surfaces and evaluation challenges [3], [4]; Singh et al. [5] demonstrated real-time burglar reporting, and Gassais et al. [6] presented host-based IoT intrusion detection. FaceNet [14] and ArcFace [15] established influential deep-learning approaches to discriminative facial representation. This work underscores the importance of diverse data, identity-separated evaluation, privacy, false-alarm control, and inference latency.

Gap analysis: Prior work covers IoT and edge architectures [1], [2], [9], [10], LPG monitoring and booking [8], [11], water-tank control [7], [12], [13], and network, physical, or camera-based intrusion detection [3]–[6], [14], [15]. Evidence nevertheless remains fragmented across application-specific prototypes. Aashraya combines gas, water, and physical-intrusion functions and evaluates resource-management, classification, and communication metrics under the stated test conditions.

III. METHODOLOGY

The Aashraya system comprises three subsystems—gas, water, and security—connected to an IoT gateway and mobile application (Figure 1). Each subsystem uses sensors, actuators, and an edge controller. Gas and water sensors feed separate ESP32-based controllers running custom firmware, while the intrusion module uses an on-site Raspberry Pi and camera. Data are communicated through MQTT, following the publish/subscribe messaging model standardized by OASIS [16], to a central broker and relayed to the smartphone application for monitoring and control. The application stores user and cylinder details and allows for configuration of thresholds and notifications.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

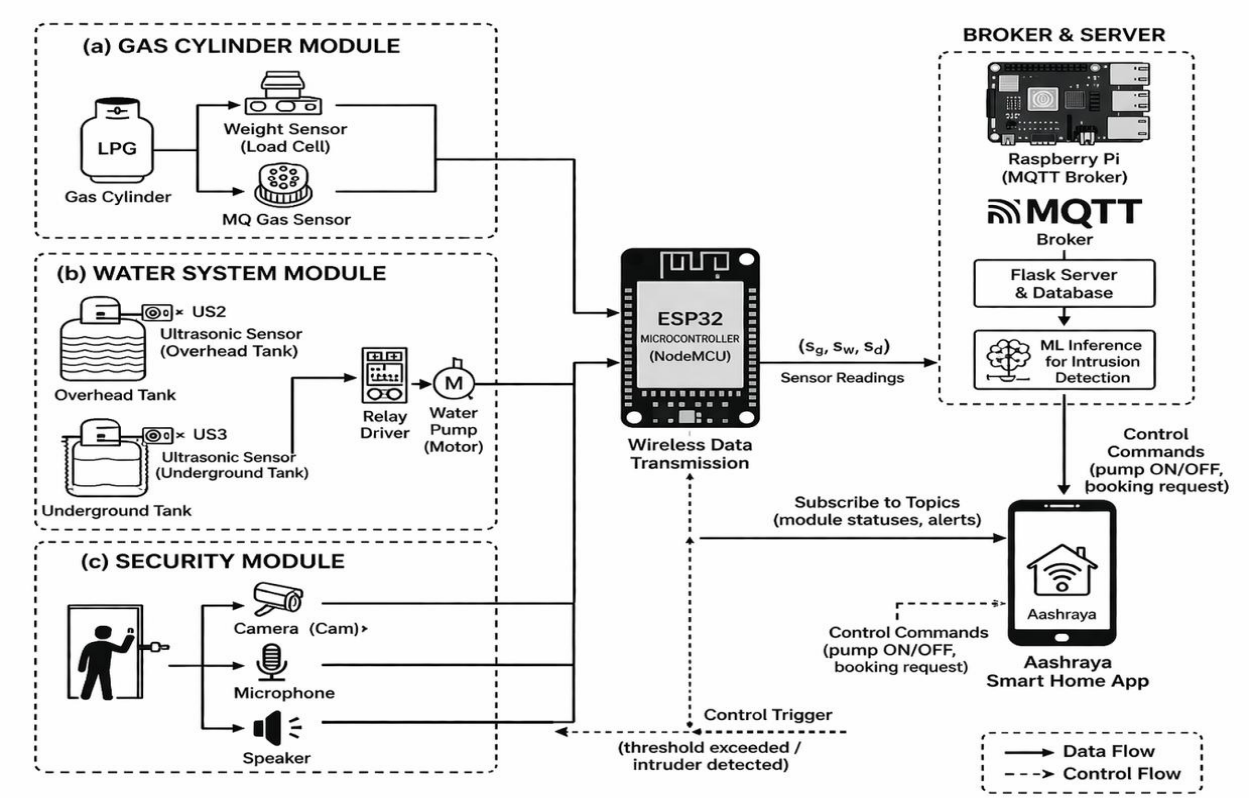

Figure.1. Proposed Aashraya smart home architecture

Figure 1 shows (a) the gas-cylinder module with a load cell and gas sensor, (b) the water module with ultrasonic sensors for the underground and overhead tanks plus a relay-driven pump, and (c) the security module with a camera, microphone, and speaker. The gas and water sensors connect to ESP32 controllers for wireless data transmission. A Raspberry Pi hosts the MQTT broker, Flask application service, database, and intrusion-inference process. The smartphone application subscribes to status and alert topics and sends control commands, including pump and cylinder-reorder requests. Solid arrows indicate data flow; dashed arrows indicate control flow.

Each subsystem is modeled mathematically as follows. Let C(t) denote the remaining gas-cylinder weight at time t and let $u_g(t)$ denote the instantaneous consumption rate. The cylinder-state model is:

$$C'^{(t)} = -u_g(t) \qquad (1)$$

where $u_g(t)$ is measured in kilograms per hour from the calibrated load cell. A safety threshold of C_{thresh} (for example, 5 kg) is defined. When C(t) reaches the threshold, or when the predicted remaining time $C(t)/u_g(t)$ falls below the configured reorder interval, the system initiates a reorder request. The prediction uses an exponentially smoothed estimate of $u_g(t)$ with a smoothing parameter α.

For the water subsystem, let $w_o(t)$ and $w_u(t)$ denote the volumes in the overhead and underground tanks, respectively. The binary pump-control signal $u_p(t) \in \{0, 1\}$ actuates the motor. The water-level dynamics are:

$$w_o'^{(t)} = \eta_p\, u_p(t) - d_o(t), \quad w_u'^{(t)} = -\eta_p\, u_p(t) + D_{in}(t) \qquad (2)$$

where η_p is the pump flow constant, $d_o(t)$ is household outflow, and $D_{in}(t)$ is the inflow from the mains. The model enforces $0 \le w_o \le W_o^{max}$ and $0 \le w_u \le W_u^{max}$. The pump turns when the overhead-tank level falls below W_o^{min} and turns

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

off when it reaches W_o^{min} + Δ; the hysteresis band Δ prevents rapid cycling. The pump is also inhibited when the underground tank falls below its minimum safe level of W_u^{min}.

The intrusion-detection module is modeled as a binary classification problem. Let $x \in R^n$ represent the feature vector produced from a door-camera image. The probability that the image contains an intruder is modeled by the logistic output:

$$P_\theta(\text{intruder}|\mathbf{x}) = \sigma(\theta^\top \mathbf{x}) = \frac{1}{1 + \exp(-\theta^\top \mathbf{x})} \qquad (3)$$

where θ denotes the learned classifier parameters. The model is trained by minimizing binary cross-entropy over labeled images of authorized residents and unknown visitors:

$$L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i \ln(p_i) + (1 - y_i)\ln(1 - p_i)] \qquad (4)$$

where p_i is the predicted intrusion probability for sample i and $y_i \in \{0, 1\}$ is the ground-truth label (0 = authorized; 1 = intruder). Stochastic gradient descent is used to estimate θ during offline training.

Overall, a composite objective function quantifies the reported system-level cost. Let J combine pump energy consumption, gas-outage duration, and false security alerts:

$$J = \alpha E_{pump} + \beta T_{gas\_out} + \gamma FP_{sec} \qquad (5)$$

where E_{pump} is cumulative pump energy, $T_{gas\ out}$ is the time spent without usable gas, and FP_{sec} is the number of false security alerts. The weights α, β, and γ represent the relative priorities assigned to these terms. Equations (1) and (2) update the gas and water states; Equation (3) is evaluated during intrusion inference; and Equation (4) is minimized during offline classifier training. Equation (5) is used only as an evaluation framework for the trade-off among energy use, gas-outage duration, and false alerts; the implementation does not claim a separate numerical optimization of J.

Table 1 lists the principal symbols and parameters. Physical constants, including pump flow rate, were taken from component specifications. Control thresholds and the smoothing parameter were tuned empirically as described in the Experimental Setup section. The architecture uses MQTT messaging and a REST API for application communication.

Table.1. Nomenclature and parameters

Symbol & Parameter	Description
C(t)	Remaining LPG cylinder weight at time t (kg)
C_{thresh}	Minimum safe gas weight threshold (kg)
$u_g(t)$	Gas usage rate (kg/h) measured via load cell
$w_o(t), w_u(t)$	Water volumes in overhead (o) and underground (u) tanks (liters)
W_o^{min}, W_u^{min}	Minimum desired water levels in tanks (liters)
η_p	Pump flow rate constant (L/h)
$u_p(t) \in \{0, 1\}$	Pump actuation signal (1 = on; 0 = off)
$d_o(t), D_{in}(t)$	Water outflow demand and inflow supply (L/h)
x,θ	x: Feature vector of the intrusion image; θ: learned parameter vector of the classifier.
y_i	Ground-truth label for sample i (0=authorized, 1=intruder)
α, β, γ	Objective weighting factors for pump energy, gas downtime, false positives

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

IV. EXPERIMENTAL SETUP

All modules were implemented on commodity IoT hardware. The gas and water sensors—a load cell and ultrasonic transducers—interface with ESP32 development boards programmed in the Arduino IDE. A USB camera connects to a Raspberry Pi 4 with 4 GB of RAM. Mosquitto 2.0 provides the MQTT broker, MariaDB stores sensor and event data, and the Android application communicates over Wi-Fi. SQLite is used separately for local application data such as customer identifiers and contact details.

Typical household usage profiles were simulated to generate resource data. Daily gas consumption was sampled from a bounded profile with a mean of 0.5 kg/day, and refill events were modeled over 30-day cycles. Water demand followed a diurnal pattern with morning and evening peaks and an average of 500 L/day. The intrusion dataset contained 700 images: 500 authorized-resident images and 200 unknown-visitor images. The data were partitioned into 80% training/validation and 20% testing, yielding a 140-image test set; the exact training/validation split was not reported. The CNN comprised two convolutional layers and one dense layer and was trained for 50 epochs with a learning rate of 0.001 and a batch size of 32. The source dataset was not identified by name, which limits reproducibility.

The experimental procedure is summarized in Figure 2. The IoT devices continuously publish sensor readings to the MQTT broker, and a Python service logs the readings for analysis. The Raspberry Pi performs camera inference; images classified as intrusions trigger a user alert through a Telegram bot. Resource-management performance was compared with manual control and static-threshold automation. The manual condition required users to initiate gas reordering, operate the pump, and monitor the door camera. The static-threshold condition used raw thresholds without consumption forecasting. Each reported scenario was repeated for 10 trials.

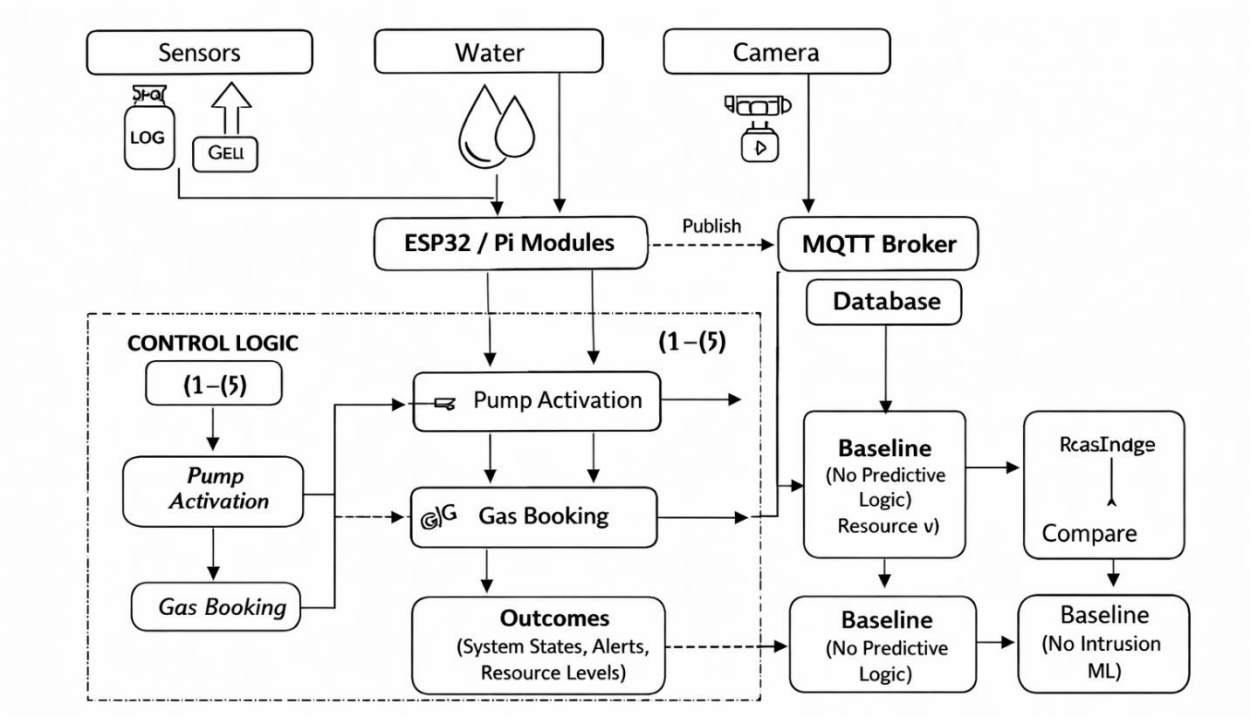

Figure.2. Experimental workflow

The workflow begins with gas, water, and camera inputs. ESP32 and Raspberry Pi nodes acquire the data and publish relevant measurements to the MQTT broker and database. Gas-reorder forecasting, water-level hysteresis, and intrusion classification produce control decisions, alerts, and logged outcomes. Baseline configurations are evaluated through the same measurement path so that gas-outage duration, pump behavior, classification performance, and communication latency can be compared consistently. Classification accuracy is computed as:

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (6)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Precision and recall are calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \qquad (7)$$

In addition, gas-outage duration, pump-control errors, water shortage incidents, false intrusion alerts, pump energy, and communication latency were tracked. MQTT latency was calculated from timestamps at publication and database receipt. CPU utilization and message throughput were observed on the ESP32 and Raspberry Pi nodes. Experiments used a single Wi-Fi network and configurations of up to 10 sensors publishing at 1 Hz. Table 2 summarizes the reported setup. Of the 700 intrusion images, 140 were reserved for testing and 560 for training and validation. Because identity-level metadata and partition details were not reported, the manuscript cannot confirm that images from the same person were restricted to a single subset; possible identity leakage is therefore an evaluation of limitation.

Table.2. Experimental configurations and datasets

Parameter	**Value / Source**
IoT Devices	2× ESP32 (gas, water), 1× Raspberry Pi
Communication	MQTT (Mosquitto v2.0), Wi-Fi (802.11g)
Gas Consumption Profile	0.2–1.0 kg/day (bounded profile; mean 0.5 kg/day)
Water Demand Profile	300–800 L/day (diurnal pattern; mean 500 L/day)
Intrusion Dataset	700 images: 560 training/validation (exact split not reported), 140 test; 500 authorized, 200 unknown
CNN Training	TensorFlow 2.11, 50 epochs, LR=0.001
Baseline Methods	Manual control; static-threshold automation
Metrics	Gas-outage duration; pump errors and energy; accuracy, precision, recall, F1; false alerts; latency

V. RESULTS

The intrusion-detection module was evaluated on the reported 140-image test set. At a decision threshold of 0.5, the confusion matrix in Table 3 contains 96 true negatives, 37 true positives, 4 false positives, and 3 false negatives. These counts yield 95.0% accuracy, 90.2% precision, 92.5% recall, and a 91.4% F1 score. Figure 3 shows the reported ROC and training-loss plots. Because an ROC curve summarizes classifier behavior across decision thresholds [17], the AUC value displayed in the figure cannot be independently verified from a single confusion matrix without the sample-level classifier scores. The broader system evaluation also reports 4 false intrusion alerts per 1,000 h, compared with 15 in the baseline condition; the underlying event counts were not reported.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

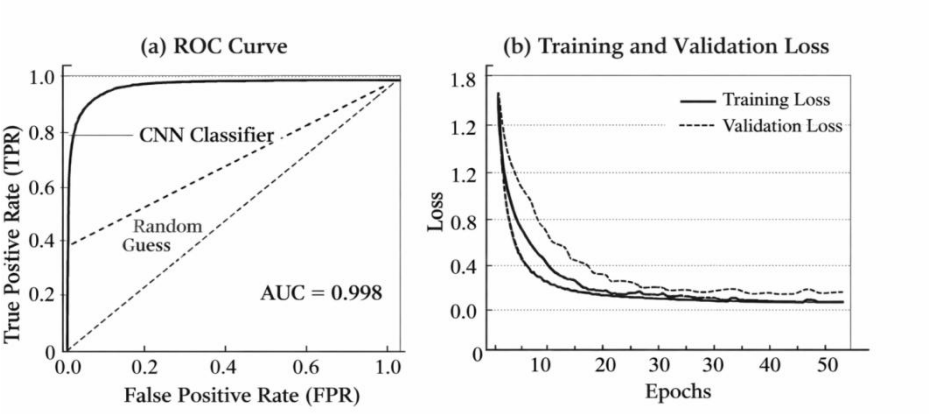

Figure.3. Intrusion detection performance. (a) Reported receiver operating characteristic (ROC) curve. (b) Reported training and validation loss across 50 epochs.

Table.3. Confusion matrix for the intrusion-detection test set (threshold = 0.5)

	Authorized (No Intruder)	**Intruder**
Predicted: No	96 (TN)	3 (FN)
Predicted: Yes	4 (FP)	37 (TP)

Note: Classification metrics were recalculated from the 140-image confusion matrix in Table 3.

For the gas-booking subsystem, average gas-outage duration decreased from 2.3 h/month in the manual baseline to 0.2 h/month with Aashraya, a 91.3% reduction. Figure 4 shows the reported booking-delay distribution, remaining gas at reorder, representative alerts, and a theft-detection simulation. The automated process-initiated reordering before projected cylinder depletion and therefore provided a multi-day replacement buffer. In one reported theft simulation, a rapid weight decrease triggered an alert and valve-closure command; this single demonstration is not a reliability estimate.

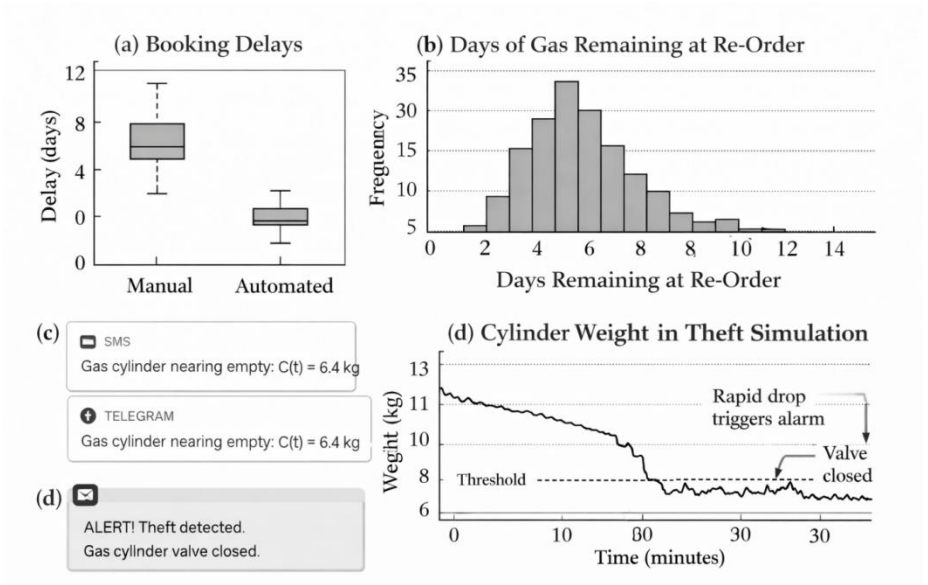

Figure.4. Gas booking and safety (a) Booking delays for the manual and automated conditions. (b) Days of gas remaining when reordering was initiated. (c) Representative reorder alerts. (d) Cylinder-weight response in the reported theft simulation

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

The booking-delay distribution in Figure 4 shows substantially earlier action under the automated condition than under manual booking. Water-management results are summarized in Figure 5. The illustrated tank-level trace remains above the lower control threshold, and the pump signal changes in response to the hysteresis limits. Across the reported trials, the pump on/off error rate decreased from 40% for the manual/static-threshold baseline to 0% for Aashraya. Average pump energy decreased from 480 to 384 Wh/day, a 20.0% reduction. The document does not provide the underlying time-series values or sensor-noise observations, so no additional quantitative claims are made about shortage, overflow, or noise robustness.

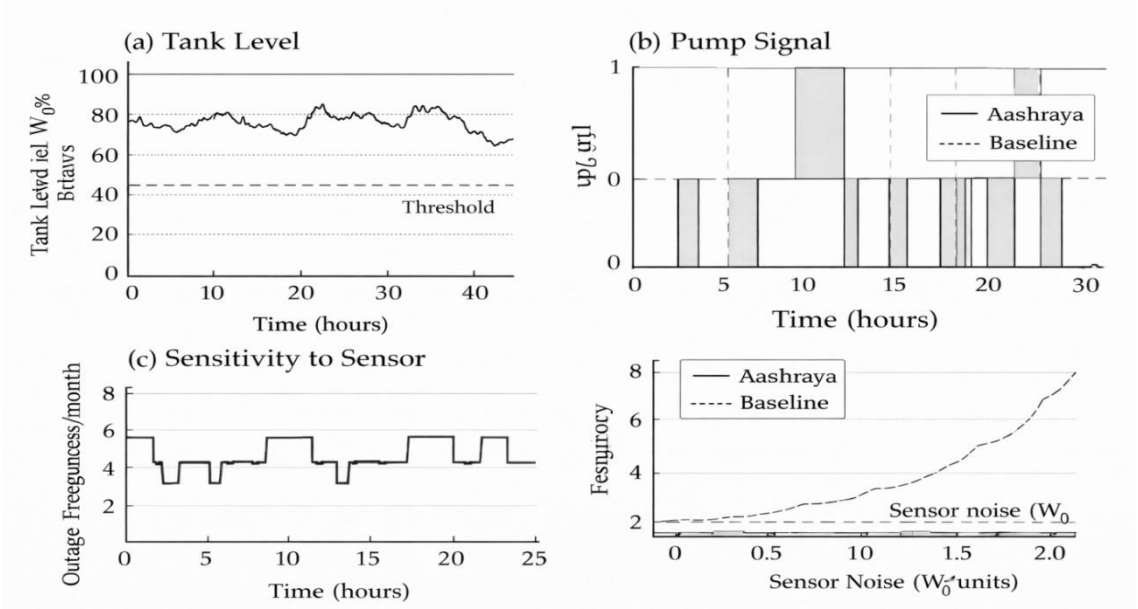

Figure.5.Water-tank control and sensitivity analysis (a) Reported overhead-tank level. (b) Reported pump-control signal. (c) Reported water-control behavior. (d) Reported sensor-noise sensitivity under hysteresis control.

Table.4. Reported performance metrics for the baseline and proposed system

Metric	Manual/Static-Threshold Baseline	Aashraya (Proposed)
Gas-outage duration (h/month)	2.3	0.2
Pump on/off error rate	40%	0%
False intrusion alerts (per 1,000 h)	15	4
Average pump energy (Wh/day)	480	384
Detection accuracy (security)	N/A	95.0%
Precision (security)	N/A	90.2%
Recall (security)	N/A	92.5%
F1 score (security)	N/A	91.4%

Note: The decrease from 2.3 to 0.2 h/month equals 91.3%, and the decrease from 480 to 384 Wh/day equals 20.0%. Security metrics were recalculated from Table 3.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

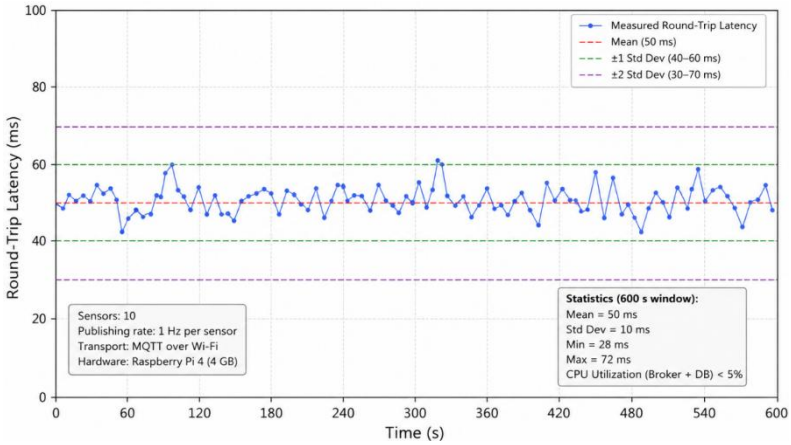

Figure.6. MQTT communication latency under multi-sensor IoT operation

With 10 sensors publishing at 1 Hz, the reported sensor-to-database MQTT round-trip latency averaged 50 ± 10 ms over the 600 s interval shown in Figure 6. Broker and database CPU utilization on the Raspberry Pi remained below 5%. These results demonstrate low latency and low edge-CPU load for the tested configuration; they do not establish scalability beyond 10 sensors or performance under different networks, publication rates, or deployment conditions.

VI. DISCUSSION

The reported results indicate improvements over the defined baseline configurations for gas, water, security, and communication functions. Gas-outage duration decreased from 2.3 to 0.2 h/month, a 91.3% reduction, and automated reordering created a multi-day replacement buffer. For water management, the pump on/off error rate decreased from 40% to 0%, while average pump energy decreased from 480 to 384 Wh/day, a 20.0% reduction. Under the evaluated conditions, these findings suggest that monitored reordering and hysteresis-based control can reduce service interruptions and unnecessary pump operation.

The intrusion classifier achieved 95.0% accuracy, 90.2% precision, 92.5% recall, and a 91.4% F1 score on the reported 140-image test set; all four values agree with the confusion matrix. However, the dataset is modest, its source is not identified, and identity-disjoint partitioning was not confirmed. The reported MQTT latency of 50 ± 10 ms and Raspberry Pi CPU utilization below 5% indicate responsive operation in the tested 10-sensor configuration, but they should not be interpreted as comprehensive evidence of real-world security, long-term reliability, or large-scale deployment performance.

Several limitations should be considered when interpreting the reported results. First, the evaluation relied partly on simulated household resource-usage profiles and a controlled IoT testbed rather than long-duration deployment across multiple real households. Consequently, the reported gas and water results characterize the evaluated operating scenarios and should not be generalized directly to all household conditions. Second, the intrusion-detection dataset comprised 700 images, including 500 authorized-resident images and 200 unknown-visitor images, and therefore represents a relatively modest evaluation set for a security application. The absence of a larger, more diverse dataset limits conclusions regarding performance under substantially different identities, poses, lighting conditions, camera positions, and environmental conditions. Third, the present study does not provide sufficient evidence to establish long-term reliability, large-scale deployment performance, or general scalability beyond the reported multi-sensor configuration. Finally, the architecture depends on network connectivity for integrated communication and alert delivery, creating a potential point of failure. Future work should therefore evaluate the system over longer deployment periods, larger and identity-separated datasets, broader environmental conditions, and additional sensor configurations, while also investigating local fail-safe operation and privacy-preserving edge inference.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

VII. CONCLUSION

This work presented Aashraya, an integrated IoT-based smart-home framework designed to combine household resource monitoring, automated gas-cylinder management, water-level control, and video-based intrusion detection within a unified architecture. The framework combines sensor-based monitoring, MQTT communication, database services, automated control logic, and a CNN-based security module to support resource management and household safety.

The reported evaluation indicates measurable improvements over the defined baselines. Average gas-outage duration decreased from 2.3 to 0.2 h/month, a 91.3% reduction, and automated reordering provided a multi-day replacement buffer. The pump on/off error rate decreased from 40% to 0%, while average pump energy decreased from 480 to 384 Wh/day, a 20.0% reduction.

For intrusion detection, the reported 140-image test set produced 95.0% accuracy, 90.2% precision, 92.5% recall, and a 91.4% F1 score based on the confusion-matrix counts. Communication evaluation reported an MQTT sensor-to-database round-trip latency of 50 ± 10 ms under the evaluated multi-sensor configuration, with 10 sensors publishing at 1 Hz and less than 5% CPU utilization reported for the broker and database on the Raspberry Pi.

Overall, the reported findings demonstrate the feasibility of integrating resource automation, security monitoring, and low-latency IoT communication within a single smart-home platform under the evaluated conditions. However, the study relies partly on simulated resource-usage profiles, a controlled test environment, and a relatively modest image dataset. Therefore, the results should not be interpreted as comprehensive evidence of long-term reliability, general scalability, or deployment performance across diverse households.

Future work should focus on longer-duration real-world deployments, larger and more diverse intrusion-detection datasets, identity-separated evaluation protocols, broader sensor configurations, and operation under variable network conditions. Further development could also investigate local fail-safe control, privacy-preserving edge inference, and adaptive or reinforcement-learning-based control strategies for more autonomous resource management.

REFERENCES

1. Taiwo, O., & Ezugwu, A. E. (2021). Internet of Things-based intelligent smart home control system. Security and Communication Networks, 2021, Article 9928254. https://doi.org/10.1155/2021/9928254
2. Yar, H., Imran, A. S., Khan, Z. A., Sajjad, M., & Kastrati, Z. (2021). Towards smart home automation using IoT-enabled edge-computing paradigm. Sensors, 21(14), Article 4932. https://doi.org/10.3390/s21144932
3. Khraisat, A., & Alazab, A. (2021). A critical review of intrusion detection systems in the Internet of Things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity, 4, Article 18. https://doi.org/10.1186/s42400-021-00077-7
4. Touqeer, H., Zaman, S., Amin, R., Hussain, M., Al-Turjman, F., & Bilal, M. (2021). Smart home security: Challenges, issues and solutions at different IoT layers. The Journal of Supercomputing, 77(12), 14053–14089. https://doi.org/10.1007/s11227-021-03825-1
5. Singh, R., Al-Khateeb, H., Ahmadi-Assalemi, G., & Epiphaniou, G. (2021). Towards an IoT community-cluster model for burglar intrusion detection and real-time reporting in smart homes. In R. Montasari, H. Jahankhani, & H. Al-Khateeb (Eds.), Challenges in the IoT and smart environments: A practitioners' guide to security, ethics and criminal threats (pp. 53–73). Springer. https://doi.org/10.1007/978-3-030-87166-6_3
6. Gassais, R., Ezzati-Jivan, N., Fernandez, J. M., Aloise, D., & Dagenais, M. R. (2020). Multi-level host-based intrusion detection system for Internet of Things. Journal of Cloud Computing, 9, Article 62. https://doi.org/10.1186/s13677-020-00206-6
7. Rojek, L., Islam, S., Hartmann, M., & Creutzburg, R. (2021). IoT-based real-time monitoring system for a smart energy house. Electronic Imaging, 2021(3), 38-1–38-10. https://doi.org/10.2352/ISSN.2470-1173.2021.3.MOBMU-038
8. Hasan, M. Z., & Ahammed, R. (2021). Application of Industry 4.0 in LPG condition monitoring and emergency systems using an IoT approach. World Journal of Engineering, 18(6), 971–984. https://doi.org/10.1108/WJE-06-2020-0218

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 1, January- February 2023 ||

DOI: 10.15680/IJCTECE.2023.0601007

9. Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095
10. 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
11. Mariselvam, V., & Siva Dharshini, M. (2021). IoT based level detection of gas for booking management using integrated sensor. Materials Today: Proceedings, 37, 789–792. https://doi.org/10.1016/j.matpr.2020.05.825
12. Gautam, G., Sharma, G., Magar, B. T., Shrestha, B., Cho, S., & Seo, C. (2021). Usage of IoT framework in water supply management for smart city in Nepal. Applied Sciences, 11(12), Article 5662. https://doi.org/10.3390/app11125662
13. Jan, F., Min-Allah, N., Saeed, S., Iqbal, S. Z., & Ahmed, R. (2022). IoT-based solutions to monitor water level, leakage, and motor control for smart water tanks. Water, 14(3), Article 309. https://doi.org/10.3390/w14030309
14. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 815–823). https://doi.org/10.1109/CVPR.2015.7298682
15. Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4690–4699). https://doi.org/10.1109/CVPR.2019.00482
16. Banks, A., Briggs, E., Borgendale, K., & Gupta, R. (Eds.). (2019). MQTT Version 5.0. OASIS Standard. https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html
17. Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010