Supervised Machine Learning Based Behavioral Pattern Recognition for Autonomous Non-Invasive Smart Switch Actuation Using a Fingerbot System

Siti Aisyah Anas*, Nur Amirah Syuhada Mohd Azlini, Anuar Jaafar, Noor Shahida Mohd Kasim, Abd Majid Darsono, Haziezol Helmi Mohd Yusof

Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer, Universiti Teknikal,Malaysia Melaka, Jalan Hang Tuah Jaya, 76100 Durian Tunggal, Melaka

***Corresponding Author**

DOI: https://doi.org/10.47772/IJRISS.2026.100600125

Received: 18 May 2026; Accepted: 24 May 2026; Published: 18 June 2026

ABSTRACT

The proliferation of smart home technologies has intensified demand for adaptive, low-cost automation solutions deployable without altering existing electrical infrastructure. This paper presents an AI-driven Fingerbot, a non-invasive electromechanical actuator that physically operates conventional wall switches through a micro servo motor governed by an ESP32-C3 microcontroller. Unlike solutions that require electrical rewiring or hardware replacement, the device mounts externally on standard switches and automates their operation through learned behavioral patterns. Three supervised classifiers; Logistic Regression, Decision Tree, and Random Forest were trained on 1,440 time-stamped user interaction records capturing hour, minute, day of week, and weekend indicator features to predict binary switch states. The Decision Tree classifier achieved 99.7% accuracy, 99.6% precision, and 99.4% recall. Random Forest recorded 97.1% accuracy with the highest precision of 99.8%, while Logistic Regression served as a linear baseline at 87.9% accuracy. Response time evaluation across three control modes showed that physical switch interaction averaged 311.67 ms (maximum 400 ms), Wi-Fi virtual control averaged 2,597.33 ms (maximum 3,750 ms), and mobile hotspot control averaged 3,257.67 ms (maximum 5,750 ms). The embedded AI activation logic commenced switch-on at approximately between 18:35 to 18:45 and enforced mandatory deactivation at 23:59 to minimize unnecessary energy consumption. Monthly energy monitoring of a 25 W test load yielded figures ranging from 121.80 kWh to 143.53 kWh across three consecutive months. The results confirm that integrating lightweight machine learning with low-cost actuator hardware constitutes a practical and cost-effective pathway for retrofit smart home automation.

Keywords: Smart home automation; Fingerbot actuator; Supervised machine learning; IoT; ESP32-C3; Behavioral pattern learning

INTRODUCTION

Residential energy management and intelligent control of household appliances have become central concerns in the design of modern smart home systems. Conventional wall switches remain entirely passive, user-operated devices that depend on physical human interaction for every state change. The integration of Internet of Things (IoT) platforms has partially addressed this limitation by enabling remote and scheduled appliance control [1][2]. However, most of the deployed systems rely on static schedules, manually defined rules, or reactive commands that cannot dynamically adapt to evolving user routines. This reactive nature places a continuous cognitive burden on users who must remember to program or update automation schedules as their habits change.

The gap between available AI technologies and their practical deployment in residential switching contexts is considerable. Most commercially available smart switch solutions require hardware replacement or permanent modification of electrical wiring which introduced barriers that significantly restrict adoption in rental properties, heritage buildings, and environments where electrical alterations are regulated or prohibited [3][4]. Research consistently identifies installation complexity and upfront cost as primary deterrents to widespread smart switch uptake [5][6]. Non-invasive mechanical actuators that interact with existing switches through physical pressing, rather than electrical integration, represent a compelling but underexplored design direction that eliminates both barriers simultaneously.

Machine learning has demonstrated feasibility for learning temporal behavioral patterns from compact, time-series datasets collected in home environments [5][6]. Tree-based ensemble classifiers consistently outperform linear models on tabular appliance activation data [7][8], yet their integration into low-cost, non-invasive actuator-based switching systems has received limited scholarly attention. Commercial *Fingerbot* products (Figure 1) currently depend on Bluetooth connectivity and pre-programmed schedules, offering no adaptive learning capability. The research described here addresses this gap by developing a custom AI-driven *Fingerbot* prototype based on an ESP32-C3 microcontroller and a micro servo motor, combined with supervised machine learning classifiers trained on historical switch interaction data, enabling autonomous pattern-based operation while preserving manual override for user confidence and safety [9].

Figure 1. Commercial *Fingerbot* product

The research objectives are to (i) design and fabricate a non-invasive mechanical Fingerbot actuator capable of reliably operating standard wall switches without electrical modification, (ii) to develop and evaluate supervised machine learning classifiers for predicting binary switch states from time-based user interaction features; and (iii) to assess the system's mechanical response characteristics and AI prediction accuracy through controlled experimental testing, including an analysis of energy consumption implications across three months of representative operation.

Related Work

IoT-Based Smart Switch Systems

The development of IoT-based smart switching systems has advanced considerably, with Wi-Fi and Bluetooth emerging as dominant communication paradigms for residential control. ESP32-based implementations have attracted considerable research interest owing to the microcontroller's dual-mode wireless capability, rich peripheral support, and low unit cost [10]. Lawrence and Soomro confirmed that ESP32-based smart switch systems reliably control household appliances via web-based interfaces [11]; Kholik et al. validated remote appliance monitoring and control through mobile applications on the same platform [4]. García-Vázquez et al. proposed an E-Switch architecture that logged usage data alongside remote-control functionality, explicitly positioning data collection as a precursor to AI-driven automation [12], a design philosophy that directly motivates the behavioral learning approach in the current work. Kumar et al. demonstrated that hardware-level fallback control substantially improves reliability during network outages [13], and Manikannan and Gopal identified unintuitive interfaces as a persistent adoption barrier among non-technical users [14].

Machine Learning for Behavioral Prediction

Yang et al. demonstrated that ML-based analysis of historical appliance usage could optimize energy scheduling and predict consumption patterns with accuracy sufficient for autonomous control [5]. Almusaed et al. showed that behavior prediction integrated into smart home design improved operational efficiency and occupant comfort across real-world deployments [15]. Gad et al.'s EL-HARP framework confirmed that ensemble tree methods consistently outperform linear classifiers on structured temporal behavioral data in smart home environments [7], a finding corroborated by Park and Kim for appliance state prediction from user behavior models [8]. Chen et al. established that lightweight edge-deployed models can match cloud counterparts on structured IoT datasets while eliminating communication overhead and inference latency [16]. Running trained classifiers directly on microcontroller firmware, without relying on external libraries, has likewise been validated in recent edge intelligence literature [6] and is applied here for the Logistic Regression model.

Non-Invasive Actuation and User Acceptance

Singh and Kumar confirmed that servo-driven actuators can reliably operate conventional switches without electrical modification, validating the mechanical feasibility of the non-invasive approach [3]. User acceptance literature consistently identifies manual override capability as a strong predictor of adoption intention, as it addresses concerns about unintended activations and system failure modes [9][13]. Systems that combine non-invasive mechanical actuation with autonomous behavioral learning from minimal data remain largely absent from the literature, and bridging this gap is the primary contribution of the present work.

METHODOLOGY

System Architecture

The proposed system integrates three subsystems: a mechanical actuation unit, an electronic control and communication platform, and a software layer for data collection, model training, and real-time embedded inference. An ESP32-C3 microcontroller manages PWM servo control signals, physical switch input monitoring via GPIO interrupt, Wi-Fi communication through the MQTT publish-subscribe protocol, and real-time embedded classification. A 3.7 V Li-Po battery, TP4056 charging module, and MT3608 boost converter supply a stable 5 V rail to the microcontroller and MG90S micro servo motor. User interaction is supported through three parallel control modes: a local hardware rocker switch connected to GPIO2, an MQTT-based web dashboard accessible over Wi-Fi, and autonomous AI-driven activation governed by the embedded classifier. This multi-modal architecture ensures operational continuity under network degradation, addressing a recognized limitation of purely network-dependent smart home systems [13].

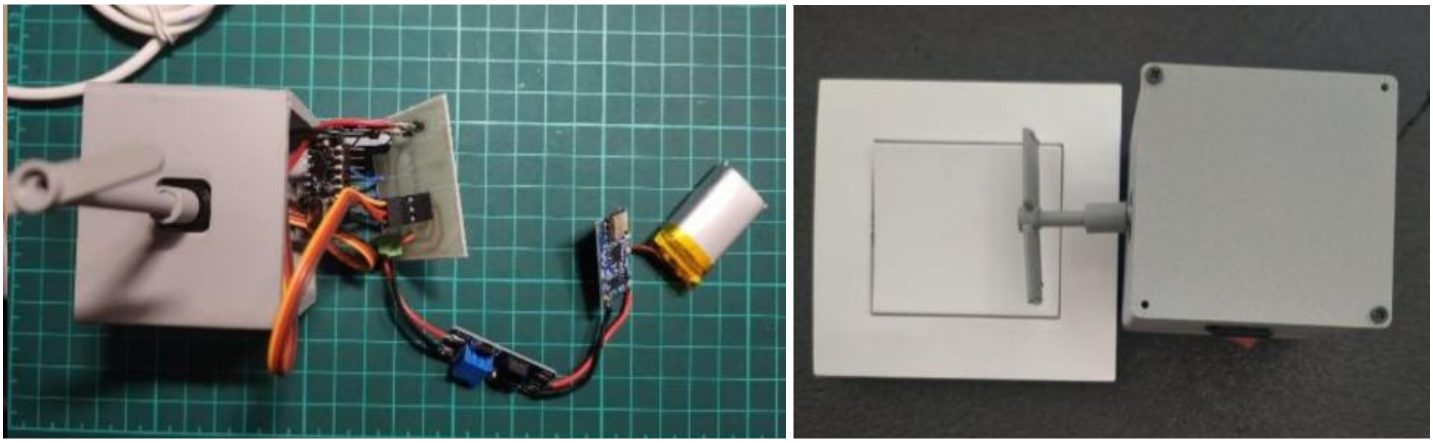

Figure 2. System prototype

Mechanical Design and Fabrication

The Fingerbot enclosure was modeled in Autodesk Fusion 360 with dimensions determined by the target mounting surface and servo motor geometry. The primary body measures 56 mm × 56 mm, with M2 screw-hole

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VI June 2026

separation of 52.424 mm, allowing secure, non-permanent attachment to standard wall switch plates without adhesive or permanent hardware alteration. Interchangeable actuator accessories accommodate rocker-type switches (outer actuator diameter: 4.60 mm) and toggle-type switches (inner bore diameter: 8.00 mm), ensuring compatibility with common residential switch formats. The completed STL model was sliced in Bambu Studio and fabricated from PLA filament on a Bambu Lab A1 printer with automatic support generation for overhanging features and pre-print automatic bed levelling. Ventilation slots around the MT3608 and TP4056 module compartments provide passive thermal management; load testing confirmed adequate airflow to maintain all components within rated thermal specifications throughout extended operation.

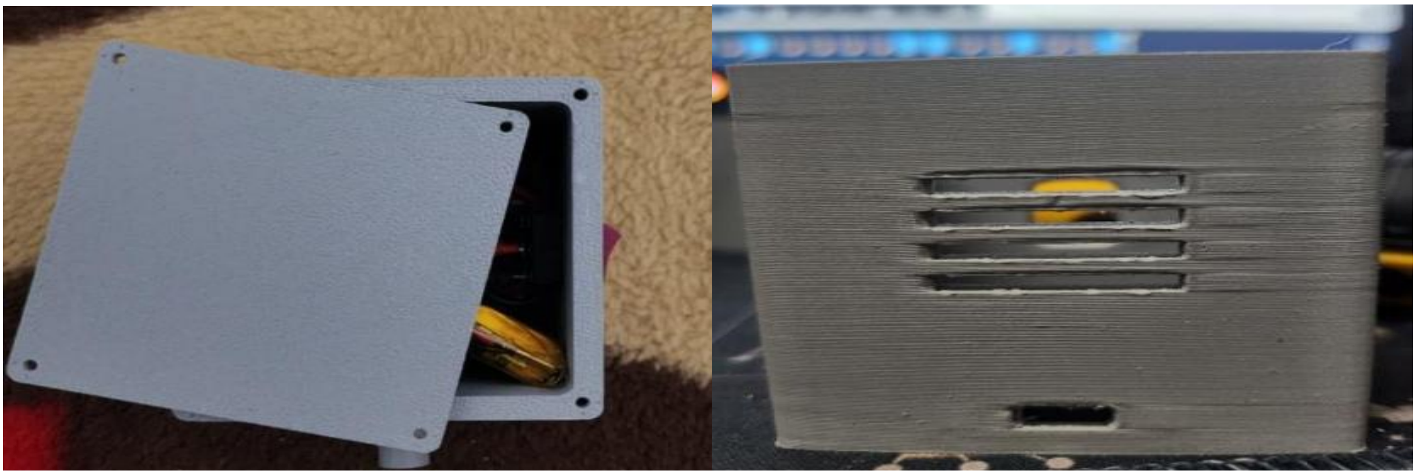

Figure 3. System enclosure

PCB Fabrication and Electronics

The circuit schematic as shown in Figure 4 was developed in EasyEDA with component placement optimized for compactness and signal integrity. Table 1 summarizes the ESP32-C3 pin assignments. PCB fabrication followed a photolithographic etching workflow: UV vacuum exposure transferred the circuit pattern to a photoresist-coated substrate, followed by four ferric chloride etching cycles for complete trace definition, conveyorized drying, precision through-hole drilling, mechanical shearing to panel-separate individual boards, and manual soldering at 350°C. All solder joints were verified for continuity and fillet quality using a digital multimeter.

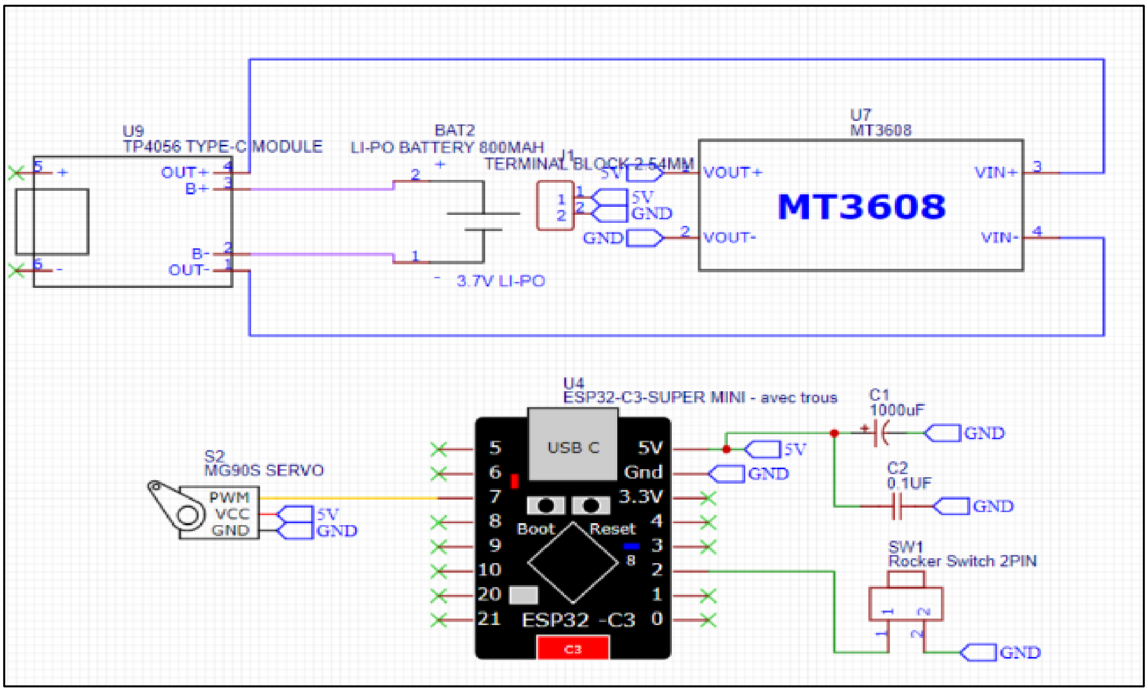

Figure 4. Circuit schematic

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VI June 2026

Table 1: ESP32-C3 Pin Assignment and Component Interconnection

ESP32-C3 Pin	Connected Component	Function
5 V	MT3608 VOUT+	Power rail
GND	MT3608 VOUT−	Common ground
GPIO2	Rocker switch	Manual input
GPIO7 (PWM)	MG90S servo motor	Actuation control

Communication, Data Collection, and Classifiers

Virtual control is implemented through MQTT, chosen for its lightweight publish-subscribe architecture and suitability for resource-constrained IoT environments [17]. The MQTT web dashboard supports bidirectional command-and-status communication and CSV data export for model training. Arduino firmware timestamped each switch activation over a representative operational period, yielding 1,440 labeled records at one-minute resolution across a 24-hour cycle. Four temporal features were extracted: *Hour* (0-23), *Minute* (0-59), *DayOfWeek* (0-6), and *IsWeekend* (binary). The binary target encodes switch state as ON (1) or OFF (0).

RESULTS

Response Time Analysis

Table 2 summarizes response time statistics across three control modes evaluated over 30 trials each. Physical switch control delivered consistently low and deterministic latency (average 311.67ms, maximum 400ms), attributable to direct GPIO interrupt handling that eliminates all network, protocol, and routing overhead. Response times ranged narrowly between 200ms and 400ms across all 30 trials; minor fluctuations reflect microcontroller interrupt service latency and rocker switch contact bounce.

Wi-Fi virtual control introduced substantially higher and more variable latency (average 2,597.33ms, maximum 3,750ms), reflecting wireless channel contention, MQTT broker processing, and network routing overhead. Mobile hotspot control produced the highest and most inconsistent latency (average 3,257.67ms, maximum 5,750ms), consistent with the higher packet loss rates and signal instability inherent in mobile data connections. Despite elevated virtual latency, all control modes remain within acceptable bounds for non-time-critical smart home applications. The physical switch's sub-400ms response serves as an essential operational fallback under network degradation, consistent with IoT reliability best practice [13].

Table 2: Response Time Statistics Across Three Control Modes (n = 30 trials)

Control Mode	Average Time (ms)	Maximum Delay (ms)
Physical Switch	311.67	400
Virtual – Wi-Fi	2,597.33	3,750
Virtual – Mobile Hotspot	3,257.67	5,750

Machine Learning Classification Performance

Table 3 reports the complete metric set for each classifier on the held-out test partition. The Decision Tree achieved near-perfect classification, with its dominant split at ~18:45 h directly encoding the observed behavioral pattern which were the switch transitions from OFF to ON in the early evening and returns to OFF by 23:59 h. Its confusion matrix shows very few false positives or false negatives, confirming that hierarchical time-threshold rules faithfully represent the switching logic present in the data. Logistic Regression achieved 87.9% accuracy but a substantially lower recall of 0.741, indicating that its linear decision boundary cannot fully capture the step-like ON/OFF transition near 18:35 h. Approximately one in four genuine activation events was misclassified as OFF, a false-negative rate that would manifest as visible missed activations in autonomous lighting control. Random Forest attained 97.1% accuracy with the highest precision (0.998), reflecting a conservative strategy that minimizes false positives (unintended activations). However, its recall of 0.906 implies that ~9.4% of genuine activation events were missed. This precision-recall trade-off is a known characteristic of ensemble majority voting on datasets with strong temporal structure [7][8] and translates in practice to roughly one missed evening activation per fortnight indicated a perceptible and frustrating user experience.

Table 3: Classification Performance of Supervised Machine Learning Models

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.879	0.833	0.741	0.784
Decision Tree	0.997	0.996	0.994	0.995
Random Forest	0.971	0.998	0.906	0.950

AI Integration Testing and Monthly Energy Analysis

End-to-end functional testing across all three control modes confirmed reliable operation. Physical rocker switch inputs triggered servo actuation within the measured latency bounds consistently across all 30 trials. MQTT virtual control achieved bidirectional communication with status acknowledgement. Decision Tree and Random Forest firmware initiated autonomous switch-on at ~18:45 across repeated test sessions; Logistic Regression triggered marginally earlier at ~18:35, reflecting the difference between a continuous probability threshold and a discrete decision boundary. All AI control modes reliably enforced deactivation at 23:59 without exception throughout extended testing.

Table 4 presents the cumulative energy consumption and estimated electricity cost of a 25W test load over three consecutive months. October and November showed relatively stable daily consumption (4.09kWh and 4.06kWh respectively), while December exhibited a 13.9% increase in daily consumption (4.63kWh), producing the highest monthly total (143.53kWh, 28.42 MYR) and reflecting extended evening operating hours associated with the festive season. The inter-month variation of 17.8% between November (lowest, 121.80kWh) and December (highest) illustrates the sensitivity of cumulative consumption to behavioral shifts. AI-driven scheduling, which constrains activation to the ~18:35–23:59 window (~5.4h/day), limits daily operating duration compared to unscheduled user-triggered operation and thereby reduces cumulative consumption over time.

Table 4: Monthly Energy Consumption and Cost – 25 W Test Load

Month	Daily Avg. (kWh)	Monthly Total (kWh)	Cost (MYR)
October	4.09	126.79	25.10
November	4.06	121.80	24.11
December	4.63	143.53	28.42

DISCUSSION

Classifier Selection for Embedded Deployment

The classification results present a clear performance hierarchy with direct practical implications for embedded deployment. The Decision Tree achieves the optimal combination of accuracy (99.7%), recall (99.4%), and embedded feasibility. Its hierarchical rule structure translates directly into Arduino nested conditionals, requires negligible inference computation on the ESP32-C3, and eliminates the memory overhead associated with ensemble methods. By contrast, Random Forest’s 9.4% false-negative rate would translate to roughly one missed evening activation per fortnight in daily use which showed a reliability failure that manifests as an absence of expected automated behavior and progressively erodes user confidence. In autonomous lighting control, where false negatives present as nothing happening rather than an error message, users may not immediately diagnose the cause, compounding frustration. These findings align with broader comparative literature. Gad et al. confirmed that ensemble methods consistently outperform linear classifiers on activity prediction tasks in smart home environments [7], while Liu et al. showed that temporal pattern modeling yields high accuracy on appliance usage classification with structured tabular features [18]. Logistic Regression, with 74.1% recall, is insufficient as a standalone autonomous classifier but retains utility as an interpretable, computationally minimal inference engine when probabilistic output or model introspection is prioritized over binary decision accuracy.

Minimal-Data Learning and Edge Deployment Implications

A practically significant feature of the proposed approach is its reliance on a compact 1,440-record, four-feature dataset. This contrasts sharply with deep learning approaches to smart home automation, which typically require large sensor-fusion datasets and substantial computing resources for training and inference [5][7]. Decision Tree's near-perfect performance on this minimal dataset reflects the highly structured nature of domestic light switching behavior: the ON/OFF transition is representable by a small number of time thresholds that a shallow tree identifies efficiently from a single day of interaction data. Chen et al. similarly established that lightweight edge-deployed models suffice for structured IoT behavioral classification without cloud dependency [16], reinforcing the conclusion that model complexity is not a prerequisite for effective automation when the target behavior is temporally regular. For households in developing regions and rental properties where sensor-rich installations, extended data collection periods, and cloud inference pipelines are infeasible, this minimal-data edge approach constitutes a genuinely accessible and economically viable automation pathway.

Latency, Network Dependency, and Control Design

The 8.3-fold latency difference between physical (avg. 311.67ms) and Wi-Fi virtual (avg. 2,597.33ms) control quantifies the concrete cost of network-mediated interaction. Wi-Fi delays fall in a transitional zone near perceptible response thresholds for interactive control; mobile hotspot delays (up to 5,750ms) clearly exceed interactive usability bounds. These observations are consistent with García-Vázquez et al.'s findings on MQTT-based smart switch latency under real-world network conditions [12]. The physical switch's deterministic sub-400ms response through direct GPIO interrupt handling makes it an indispensable fallback mechanism, confirming that multi-modal control architectures with local physical override are a structural reliability requirement rather than a convenience feature for deployed IoT systems [13].

Energy Scheduling and Limitations

Restricting switch activation to the ~5.4h evening window substantially reduces potential daily operating duration relative to unscheduled user-triggered operation, which can span 8–16 hours under typical residential occupancy. Yang et al. demonstrated that ML-based behavioral scheduling reduces appliance energy waste by optimizing activation timing against observed usage distributions [5]; the present system operationalizes this principle at the single-switch level using learned rather than manually specified timing, requiring no user configuration beyond the initial data collection phase. The 17.8% inter-month energy variation underscores the limitation of a static trained model: gradual seasonal shifts in user behavior cannot be accommodated without retraining. Key limitations of the current implementation include a single-day training dataset that limits generalization to irregular or multi-occupant usage patterns; MG90S torque constraints (~1.8 kg-cm at 4.8 V) that may be insufficient for stiff or aged wall switches in diverse field environments; and the absence of TLS encryption on the MQTT channel, which would be mandatory in any production deployment to prevent unauthorized control and data interception.

CONCLUSION AND FUTURE DIRECTIONS

This paper has demonstrated an AI-driven Fingerbot system that automates conventional wall switches through non-invasive mechanical actuation combined with supervised machine learning trained on compact temporal interaction data, requiring no electrical modification and no cloud infrastructure. The Decision Tree classifier achieved 99.7% accuracy and 99.4% recalled the optimal balance between classification fidelity and embedded deployment feasibility on the ESP32-C3. It learned evening activation threshold (~18:45h) and mandatory 23:59 h deactivation enable autonomous, energy-aware operation. Random Forest achieved the highest precision (99.8%) but its 9.4% false-negative rate constitutes an unacceptable reliability gap for autonomous lighting control. Logistic Regression provides a computationally minimal embedded baseline through hardcoded sigmoid inference, albeit with insufficient recall for reliable autonomous activation. Physical switch control delivered deterministic sub-400ms latency as an essential network-independent fallback, while MQTT virtual control extended remote accessibility at the cost of network-dependent delays. Monthly energy monitoring confirmed that behavioral scheduling measurably constrains cumulative consumption, with 17.8% seasonal variation motivating future incorporation of online learning capability. Together, these results demonstrate that integrating lightweight classification with low-cost servo actuation constitutes a practical, affordable, and broadly deployable pathway for retrofit smart home automation.

Future work should integrate PIR and ambient light sensors to augment temporal features with real-time occupancy context, enabling the classifier to distinguish between scheduled activations and unoccupied periods. Upgrading the actuation mechanism to a higher-torque servo (e.g., MG996R) would extend deployment compatibility to stiffer switch hardware. Incorporating online learning algorithms that update model parameters incrementally from new interaction data would enable adaptation to gradual seasonal behavioral shifts without manual retraining cycles. Scaling the architecture to coordinate multiple *Fingerbot* units across different rooms would enable whole-home energy management scenarios. TLS-secured MQTT communication, alternative enclosure materials such as PETG or ABS for improved thermal and mechanical robustness, and ESP32 deep-sleep power management modes for extended battery life represent essential engineering refinements toward a production-grade, secure, and energy-autonomous deployment.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Fakulti Teknologi dan Kejuruteraan Elektronik dan Komputer (FTKEK) at Universiti Teknikal Malaysia Melaka (UTeM) for their valuable support in providing the necessary facilities, information, and resources that contributed to the successful completion of this research. Special appreciation is also extended to colleagues and participants who generously contributed their time and feedback during the study.

REFERENCES

1. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," IEEE Trans. Ind. Informat., vol. 10, no. 4, pp. 2233-2243, 2014.
2. A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart homes: A comprehensive survey," IEEE Commun. Surveys Tuts., vol. 16, no. 1, pp. 1190-1220, 2014.
3. R. Singh and S. Kumar, "Non-intrusive smart home automation using actuator-based IoT devices," IEEE Sensors J., vol. 23, pp. 8421-8430, 2023.
4. M. A. Kholik, N. A. Fajrin, and S. F. Budi, "Design and implementation of a smart home system with the Internet of Things (IoT) using ESP32," Inspirasi J. Teknol. Inf. Komunikasi, vol. 13, no. 1, pp. 22-30, 2023.
5. Y. Yang, W. Wang, P. Peng, J. Wang, and Y. Zhang, "User behavior learning for smart home energy management using machine learning," IEEE Access, vol. 9, pp. 118452-118463, 2021.
6. A. S. Ali, M. Ghani, K. Talib, and A. S. Ahmad, "Intelligent home automation using machine learning and IoT," IEEE Access, vol. 10, pp. 55840-55852, 2022.
7. M. M. Gad, W. Gad, T. Abdelkader, and K. Naik, "Personalized smart home automation using machine learning: Predicting user activities," Sensors, vol. 25, no. 19, p. 6082, 2025, doi: 10.3390/s25196082.
8. J. Park and H. Kim, "Predictive control of smart home devices based on user behavior modeling," IEEE Trans. Consum. Electron., vol. 68, no. 3, pp. 289-298, 2022.
9. S. Alotaibi and M. A. Hossain, "User trust and acceptance of intelligent home automation systems with manual override," IEEE Access, vol. 8, pp. 168345-168356, 2020.
10. D. Chikurtev, B. Karkov, and E. Blagoeva, "An overview of ESP32 microcontroller application in embedded systems, robotics and IoT," Springer Proc. Math. Stat., vol. 514, 2026.
11. S. O. Lawrence and D. M. Soomro, "IoT-based smart home switch system using the ESP32 microcontroller," Evol. Electr. Electron. Eng., vol. 3, no. 2, pp. 856-862, 2022.
12. H. A. Garcia-Vazquez et al., "Design and implementation of the E-Switch for a smart home," Sensors, vol. 21, no. 11, pp. 1-17, 2021.

ISSN No. 2454-6186 | DOI: 10.47772/IJRISS | Volume X Issue VI June 2026

13. P. Kumar, D. S. K. Singh, M. S. H. A. Mishra, and R. H. S. P. Sharma, "Design and fabrication of IoT switch with manual override," IOP Conf. Ser.: Mater. Sci. Eng., 2021.
14. P. Manikannan and M. S. Gopal, "IoT enabled smart switch with user-friendly electrical interfacing," in Proc. ICSTSN, 2022, pp. 1-6.
15. A. Almusaed, I. Yitmen, and A. A. Al-Amri, "Enhancing smart home design with AI models: A case study of living spaces implementation review," Energies, vol. 16, no. 6, p. 2636, 2023.
16. L. Chen, Y. Zhang, and H. Wang, "Lightweight machine learning models for edge-based smart home applications," IEEE Internet Things J., vol. 9, no. 7, pp. 5150-5160, 2022.
17. M. Salama and B. Raslen, "MQTT in action: Building reliable and scalable home automation systems," Sakarya Univ. J. Comput. Inf. Sci., vol. 7, no. 3, pp. 494-509, 2024.
18. H. Liu, Y. Wang, and X. Zhang, "Time-series prediction of appliance usage for smart homes using deep learning," IEEE Access, vol. 11, pp. 10321-10332, 2023.
19. W. A. Abd Elkhalik, "AI-driven smart homes: Challenges and opportunities," J. Intell. Syst. Internet Things, vol. 8, no. 2, pp. 54-62, 2023.
20. P. R. Kumar and S. K. Panda, "Data-driven decision making in IoT-based smart home automation," J. Ambient Intell. Humaniz. Comput., vol. 12, no. 4, pp. 3891-3902, 2021.
21. R. A. Rahman, M. S. H. Shah, and G. M. A. Alam, "AI-enabled smart home systems: Architecture, challenges, and future directions," IEEE Access, vol. 12, pp. 15011-15025, 2024.
22. A. R. K. Singh and K. Singh, "IoT based smart switchboard technology for smart home," J. Univ. Shanghai Sci. Technol., vol. 23, no. 9, pp. 724-737, 2021.
23. T. M. Thomas et al., "Analysis of challenges and solutions of IoT in smart grids using AI and machine learning techniques: A review," Electronics, vol. 12, no. 1, p. 242, 2023.