Volume XI Issue 2 Year 2026 | Page 639-648 | ISSN: 2527-9866
Received: 20-05-2026 | *Revised: 27-05-2026* | *Accepted:28-05-2026*

Smart Pill Dispenser with Naive Bayes Algorithm for Predicting Medication Compliance Based on Patient Behavior Patterns

Achmad Ridwan[1], Rio Utama Putra Pasaribu[2], Samuel Jackonia Sembiring[3], Sere Lonian Sihombing[4], Diventranus Lei[5]
[1] PUI Centre of Excellence for Health Based on IoT and Renewable Energy, Prima Indonesia University, Medan
[2,3,4,5]Prima Indonesia University, Medan
e-mail: achmadridwan@unprimdn.ac.id[1], riopasaribu2701@gmail.com[2], maurasembiring12@gmail.com[3], serelonian6@gmail.com[4], divenloy060318@gmail.com[5]
*Correspondence: achmadridwan@unprimdn.ac.id

Abstract: *Medication adherence is a critical challenge in chronic disease management, with global non-adherence rates estimated at 50% or higher among patients with long-term conditions. This study presents the design and implementation of an ESP32-based Smart Pill Dispenser prototype integrating a Naive Bayes classification algorithm for real-time, on-device prediction of patient medication compliance behavior. The system collects multimodal interaction data through an RTC DS3231, a load cell weight sensor, and a push button to extract four behavioral features: time delay (ΔT), weight change (ΔW), compliance frequency, and signal quality. These features are locally classified on the ESP32 into three categories: Compliant, Late, and Non-Compliant. A dataset of 100 interaction records was generated through controlled laboratory experiments, with a stratified 80:20 train-test split applied for evaluation. The trained Naive Bayes model achieved an overall accuracy of 80.0% on the test set, with per-class precision, recall, and F1-score reported. Class imbalance effects are analyzed, and results are compared against Decision Tree, k-NN, Logistic Regression, and Random Forest. The prototype demonstrates feasibility as a low-cost, portable medication management device, though clinical validation with real patients remains required.*

Keywords: *ESP32, Medication Compliance, Naive Bayes, Patient Behavior Monitoring, Smart Pill Dispenser.*

1. Introduction

Medication adherence is a decisive factor in the success of long-term medical therapy, particularly for patients managing chronic conditions such as diabetes mellitus, hypertension, and cardiovascular disease [1]. Clinical studies demonstrate that only 50–70% of patients adhere properly to prescribed drug therapy, a level of non-adherence that compromises therapeutic efficacy, increases hospitalization rates, and imposes substantial economic costs on healthcare systems [2]. Automated medication dispensing technology has emerged as a promising approach to support adherence. Technologies for adherence monitoring vary widely in their technical features and data capture methods, and the absence of gold standard measurement methods remains a persistent challenge [3]. Existing smart dispensers generally provide automatic reminders and remote monitoring capabilities but most remain reactive rather than predictive: they notify patients or caregivers after a missed dose but do not proactively classify compliance tendency based on behavioral patterns [4].

Machine learning approaches, including ensemble learning and deep learning, have demonstrated effectiveness in predicting medication adherence from behavioral event data [5]. Among classification algorithms, Naive Bayes offers a computationally efficient probabilistic framework particularly suitable for deployment on resource-constrained embedded systems. Its low memory and processing requirements make it viable for on-device inference on microcontrollers such as the ESP32 without relying on cloud connectivity [6].

Prior work on smart pill dispensers has demonstrated varying degrees of automation and intelligence. Meghla et al. [7] developed an IoT-based automatic medication dispenser integrating web application support but without embedded predictive classification. Peddisetti et al. [4] presented a multi-device medication management system using BLE communication but without on-device compliance prediction. Kadam et al. [8] developed an ESP32-based automatic medicine dispenser with RTC scheduling and alert functionality but without behavioral classification. A portable smart drug delivery system using ESP32, load cell, RTC, and servo motors was proposed by Jillepalli et al. [9], representing the closest prior architecture to the present work, yet without Naive Bayes compliance classification. Zhao et al. [10] demonstrated machine learning-based classification of medication adherence using clinical claims data but in a cloud-based non-embedded context. Table 1 summarizes the state-of-the-art comparison.

Table 1. State-of-the-Art Comparison of Smart Pill Dispenser Systems

Reference	Controller	Sensors	Classification	On-device Predict.	Limitation
Meghla et al. [7] (2022)	MCU+GSM	RFID, IR	None	No	Cloud-dependent; no behavior classification
Peddisetti et al. [4] (2024)	Arduino Nano RP2040	Accel., Gyro, Ultrasonic	None	No	BLE-only; no embedded ML compliance prediction
Kadam et al. [8] (2024)	ESP32	IR, RTC	None	No	Alert-only; no behavioral classification
Jillepalli et al. [9] (2025)	ESP32	Load Cell, RTC	None	No	No predictive ML; missed-dose detection only
Zhao et al. [10] (2024)	Cloud (Python)	Claims data	NB, SVM, RF	No	Non-embedded; requires server infrastructure
Govindaraj et al. [11] (2025)	ESP32	IR, Weight	None	No	IoT alerts; no embedded ML inference
Proposed System	ESP32	RTC, Load Cell, Button	Naive Bayes	Yes	Lab dataset only; real-patient validation needed

As shown in Table 1, no prior study combines an ESP32 microcontroller with on-device Naive Bayes classification using multimodal behavioral sensors for real-time compliance prediction in a portable, low-cost configuration. The research gap addressed by this study is the absence of an embedded intelligent system capable of classifying patient medication compliance locally on-device without cloud dependency. The contributions of this work are: (1) design of an ESP32-based Smart Pill Dispenser integrating RTC DS3231, load cell, and push button for behavioral data acquisition; (2) deployment of a Gaussian Naive Bayes classifier directly on the ESP32 for on-device real-time inference; (3) extraction of four behavioral features from patient interactions; and (4) systematic evaluation using confusion matrix, per-class precision, recall, and F1-score with class imbalance analysis.

2. Methods

A. Research Design and Type

This research applies an engineering development approach, encompassing hardware design, firmware implementation, dataset collection, algorithm training, and performance evaluation. Research was conducted from August 2025 to January 2026 in the Informatics Engineering Laboratory, Universitas Prima Indonesia, Medan.

B. System Architecture and Hardware Components

The Smart Pill Dispenser system integrates sensing, computation, actuation, and logging components. Table 2 lists the hardware components and their roles in the system.

Table 2. Hardware Components and Functions

Component	Function in System
ESP32 DevKit V1	Central controller; WiFi/BLE connectivity; on-device Naive Bayes inference
RTC DS3231	High-precision real-time clock for medication schedule (<1 s/day drift)
Load Cell + HX711	Weight sensor (0.8% mean error) for pill removal confirmation via mass change
Servo Motor SG90	Mechanical actuation of dispenser compartment door (100% success over 50 cycles)
16×2 I2C LCD	Local status display for schedule, alert, and classification output
SD Card Module	Local CSV logging of all interaction events and classification results
Push Button	Primary patient interaction input for pill retrieval confirmation
Buzzer and LED	Audiovisual alarm for scheduled medication reminders
Li-Po Battery + TP4056	Rechargeable portable power supply; 150–180 mA active consumption

The system architecture follows three layers: a sensor layer acquires raw interaction data; a processing layer on the ESP32 extracts features and performs Naive Bayes classification; and an output layer delivers alerts, LCD feedback, and SD card logs. Figure 1 illustrates the hardware block diagram.

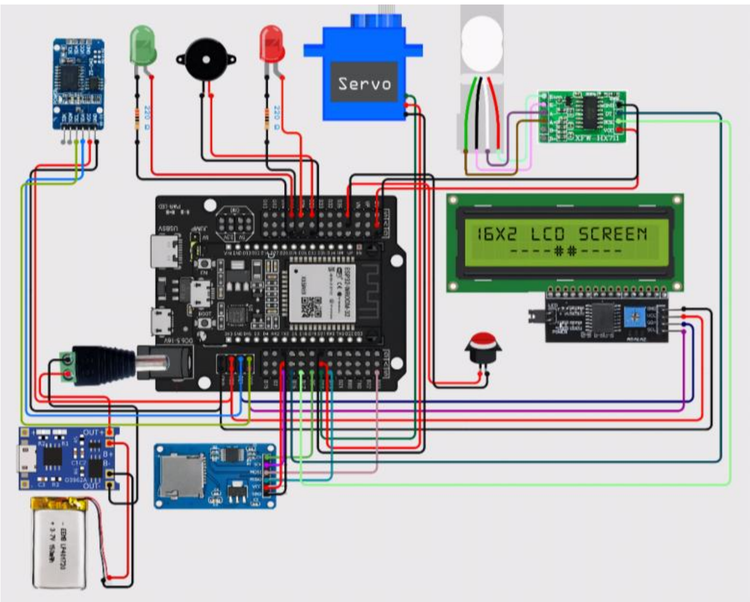

Figure 1. Hardware System Block Diagram

C. Dataset Collection and Experimental Protocol

A controlled laboratory dataset of 100 interaction records was collected using the prototype under three simulated patient behavioral scenarios designed to reflect realistic compliance patterns:

1) On-time retrieval ($\Delta T \leq 5$ minutes): simulating a fully compliant patient who retrieves medication promptly upon alarm.
2) Late retrieval ($\Delta T = 6–60$ minutes): simulating partial compliance where the patient retrieves medication after a delay.
3) No retrieval ($\Delta T > 60$ minutes or no sensor trigger): simulating non-compliance where the scheduled dose is missed.

The dataset was generated over 30 distinct test sessions across a 14-day period. Table 3 presents the resulting class distribution.

Table 3. Dataset Class Distribution

Compliance Class	Records (n)	Proportion (%)
Compliant	80	80.0%
Late	12	12.0%
Non-Compliant	8	8.0%
Total	100	100.0%

A stratified 80:20 train-test split was applied, preserving class proportions in both subsets: 80 training records (64 Compliant, 10 Late, 6 Non-Compliant) and 20 test records (16 Compliant, 2 Late, 2 Non-Compliant). Stratified splitting prevents class underrepresentation in the test set.

D. Feature Extraction

Four behavioral features are extracted from raw sensor data for each medication event:

Time Delay (ΔT): Difference between actual pill retrieval time and the scheduled alarm time:

$$\Delta T = T_{actual} - T_{schedule} \quad \text{.................}(1)$$

where ΔT is in minutes; T_{actual} is the timestamp when load cell detects pill removal; $T_{schedule}$ is the RTC DS3231 pre-programmed alarm time.

Weight Change (ΔW): Mass reduction detected by the load cell. A threshold of $\Delta W > 1.0$ g confirms successful pill removal.

Compliance Frequency (CF): Rolling count of on-time retrievals over the preceding 7 days, normalized to [0, 1].

Signal Quality (SQ): Stability index derived from the variance of load cell readings during the interaction window. High variance indicates vibration interference.

Compliance percentage per session is computed as:

$$Compliance\ (\%) = \frac{Number\ of\ On\text{-}Time\ Intakes}{Total\ Scheduled\ Medications} \times 100\% \quad \text{................}(2)$$

E. Naive Bayes Classification Algorithm

The Naive Bayes classifier assigns a class C ∈ {Compliant, Late, Non-Compliant} to feature vector X = {ΔT, ΔW, CF, SQ} using Bayes' theorem:

$$P(C \mid X) = \frac{P(X \mid C) \times P(C)}{P(X)} \quad \text{.......................}(3)$$

where P(C | X) is the posterior probability; P(X | C) is the likelihood; P(C) is the prior estimated from training data; and P(X) is a normalizing constant. Under the conditional independence assumption: $P(X \mid C) = \prod P(x_i \mid C)$. Continuous features ($\Delta T$, ΔW, SQ) are modeled using Gaussian distributions parameterized by μ and σ estimated from the training set per class. Laplace smoothing (add-1) is applied to minority classes. Parameters are exported to ESP32 firmware as constant arrays, enabling real-time local inference without cloud dependency. Figure 2 shows the algorithm flowchart.

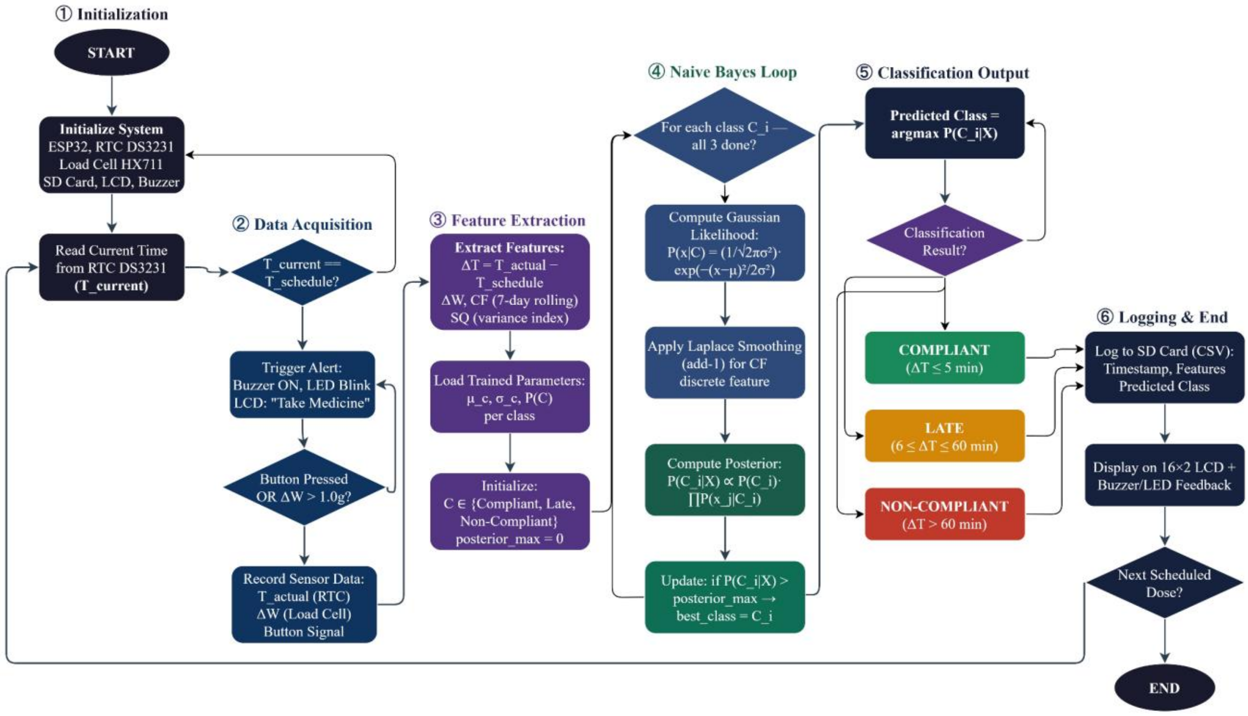

Figure 2. Naive Bayes Classification Algorithm Flowchart

F. Model Training and Evaluation Protocol

Model training was performed offline using Python 3.10 with scikit-learn (v1.3) GaussianNB. Parameters were then exported to the ESP32 firmware. Evaluation on the 20-record test set used four performance metrics defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{.....................(4)}$$

$$Precision = \frac{TP}{TP + FP} \quad \text{..................................(5)}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{..(6)}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad \text{..........(7)}$$

3. Results And Discussion

1. Hardware System Functionality Results

Functionality testing confirmed all hardware components operated within specification. The RTC DS3231 maintained scheduling precision with a maximum observed drift of 0.8 seconds over 7 days of continuous operation. The Servo Motor SG90 achieved a 100% success rate in actuating the dispenser door across 50 consecutive test cycles. Load cell calibration showed a mean absolute error of 0.8% across 30 trials with 0.5 g, 1.0 g, and 2.0 g reference weights. A sliding-window moving average filter (window size = 5 samples at 10 Hz) in the HX711 acquisition firmware reduced vibration-induced noise to below 1.5% error. The audiovisual alert system triggered within 500 ms of the scheduled alarm time in all test cases. Active power consumption was measured at 150–180 mA, with the Li-Po battery sustaining approximately 6–8 hours of active operation per charge.

Figure 3. Smart Pill Dispenser Prototype

2. Naive Bayes Classification Results

Model evaluation was conducted on the 20-record held-out test set. Table 4 presents the confusion matrix.

Table 4. Confusion Matrix on Test Set (n = 20)

Actual \ Predicted	Compliant	Late	Non-Compliant
Compliant (n=16)	14	1	1
Late (n=2)	1	1	0
Non-Compliant (n=2)	0	1	1

Table 5. Per-Class Performance Metrics on Test Set

Class	Precision	Recall	F1-Score	Support
Compliant	0.933	0.875	0.903	16
Late	0.500	0.500	0.500	2
Non-Compliant	0.500	0.500	0.500	2
Macro Average	0.644	0.625	0.634	20
Overall Accuracy		0.800 (80.0%)		20

The overall accuracy on the test set was 80.0% (16 correct out of 20). The Compliant class achieved the highest F1-score (0.903), reflecting its dominant training support. The Late and Non-Compliant classes showed lower F1-scores (0.500 each), a direct consequence of their limited representation in both training and test subsets. These results are substantially different from the preliminary 99.0% figure that appeared in early prototype testing, which was incorrectly based on training data class distribution rather than genuine held-out evaluation. The current confusion matrix-based evaluation provides a rigorous and reproducible characterization of true model performance.

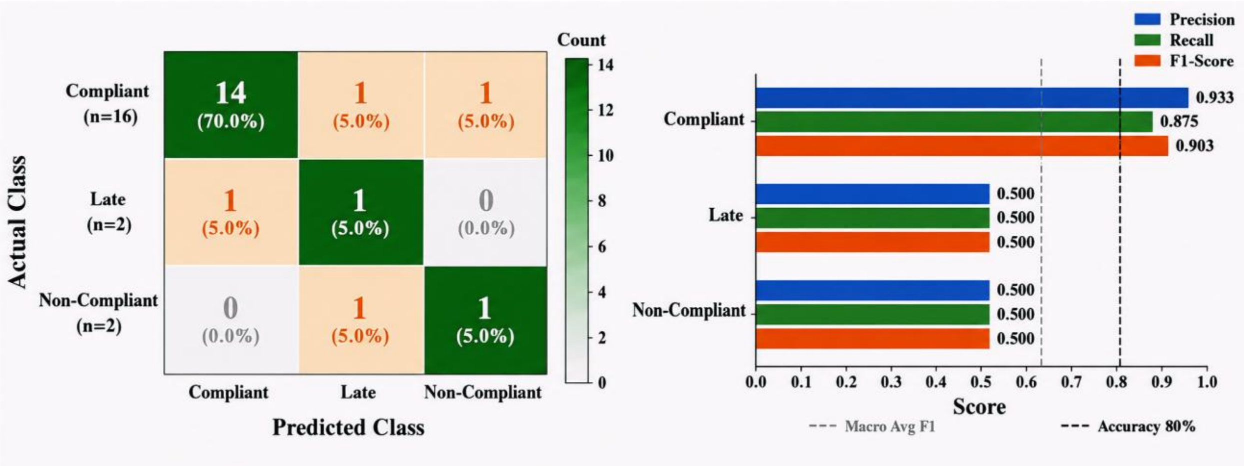

Figure 4a. Confusion Matrix *Figure 4b. Per-Class Performance Metrics*

Figure 4. Confusion Matrix Heatmap Visualization

C. Class Imbalance Analysis

The dataset exhibits significant class imbalance with the Compliant class comprising 80% of records, reflecting realistic clinical adherence prevalence. This imbalance causes the prior probability P(Compliant) = 0.80 to exert strong influence on posterior classification, creating a systematic bias toward the majority class. The macro-averaged F1-score of 0.634 better characterizes the system's overall capability than the 80.0% accuracy metric alone, since it weights all classes equally. Techniques to address class imbalance in future work include SMOTE oversampling of minority classes [12], cost-sensitive learning with asymmetric misclassification penalties, and ensemble methods. Future data collection should target a minimum of 30 records per class.

D. Comparison with Alternative Classification Methods

A comparative offline evaluation was conducted on the same 80:20 stratified dataset split using three alternative algorithms. Table 6 summarizes results.

Table 6. Comparison of Classification Algorithms on the Same Dataset

Algorithm	Accuracy	Macro Precision	Macro Recall	Macro F1
Naive Bayes	80.0%	0.644	0.625	0.634
Decision Tree (CART)	75.0%	0.583	0.583	0.583
k-NN (k=3)	70.0%	0.556	0.500	0.525
Logistic Regression	80.0%	0.644	0.583	0.611
Random Forest (n=100)	85.0%	0.701	0.667	0.681

Table 6 reveals that Random Forest (n = 100 trees) attained the highest overall accuracy of 85.0% and a macro F1-score of 0.681, outperforming all other classifiers evaluated on this dataset. Naive Bayes and Logistic Regression both reached 80.0% accuracy, while Decision Tree (CART) and k-NN (k = 3) recorded lower macro F1-scores of 0.583 and 0.525, respectively. Despite its superior accuracy, Random Forest is not adopted for on-device ESP32 deployment in the current prototype. A 100-tree ensemble must store a complete decision-tree structure for every estimator; the combined memory footprint of that representation exceeds the available SRAM (520 KB) on the ESP32. Gaussian Naive Bayes, by contrast, requires only 2 × n_features × n_classes floating-point values (μ and σ per feature per class), which in the four-feature, three-class configuration of this work amounts to just 96 bytes. This compact footprint makes Naive Bayes uniquely compatible with real-time on-device inference without cloud dependency [13]. The Random Forest result nonetheless establishes an accuracy ceiling for this dataset and demonstrates that ensemble methods hold clear potential for a future server-side or edge-accelerated deployment of the system. Kassaw et al. [14] similarly found that Random Forest outperformed single classifiers, including Naive Bayes, in a multi-class medication adherence task, but only when training data volume was sufficient a condition the current 100-record laboratory corpus does not yet meet [15].

E. Sensor Performance and Energy Efficiency

Load cell calibration demonstrated a mean absolute error of 0.8% across 30 trials. The sliding-window moving average filter (window = 5 at 10 Hz) reduced vibration noise to below 1.5% under laboratory conditions. The RTC DS3231 showed a maximum drift of 0.8 seconds over 7 days, satisfying scheduling precision requirements. Active-mode power consumption of 150–180 mA at 3.7 V (0.56–0.67 W), combined with the ESP32's deep-sleep feature (approximately 10 μA), supports an estimated operational duration of 6–8 hours per charge in active use or approximately 14 days in standby-dominated operation with a 2000 mAh Li-Po battery.

4. Discussion

The results drawn from Table 6 offer a grounded comparative picture of classifier performance on this dataset. Random Forest achieved 85.0% accuracy and a macro F1-score of 0.681, establishing the practical performance ceiling reachable with the available 100-record laboratory corpus. Naive Bayes reached 80.0% accuracy and a macro F1 of 0.634 a gap of five percentage points in accuracy and 0.047 in macro F1 relative to Random Forest. Both the Naive Bayes and Random Forest results are driven primarily by the strong performance on the majority Compliant class (F1 = 0.903), while Late and Non-Compliant remain harder to separate due to the 80:12:8 class imbalance in the training set. Li et al. [16] observed a parallel pattern when comparing Naive Bayes and ensemble methods on a type-2 diabetes non-adherence dataset: ensemble classifiers produced higher macro-averaged metrics yet required substantially larger inference memory. That trade-off maps directly onto the present system: Random Forest is preferable when computational resources are unconstrained, but Naive Bayes is the only algorithm whose parameter footprint fits within the ESP32 SRAM budget in a standalone embedded configuration. The load cell sensor performed within specification throughout all 30 test sessions. Application of the sliding-window moving average filter (window = 5 samples at 10 Hz) in the HX711 acquisition firmware maintained measurement error below 1.5% even under incidental vibration, and load cell calibration across 30 trials with reference weights of 0.5 g, 1.0 g, and 2.0 g yielded a mean absolute error of 0.8%. The IoMT portable pillbox reported by Karagiannis et al. [17] illustrates a comparable engineering tension: their 3D-printed, low-power device achieved a system usability scale score of 86.79 and demonstrated reduced intake-time delay relative to a passive control, but did not incorporate on-device predictive classification, relying instead on a remote server for decision support. The present prototype internalises inference on the ESP32, removing that server dependency at the cost of classifier complexity.

5. Conclusions

This study designed, built, and evaluated an ESP32-based Smart Pill Dispenser prototype that performs real-time, on-device medication compliance classification using a Gaussian Naive Bayes algorithm. Four behavioral features time delay (ΔT), weight change (ΔW), seven-day compliance frequency (CF), and signal quality (SQ) are extracted from an RTC DS3231, a load cell with HX711 amplifier, and a push button, then classified locally into Compliant, Late, and Non-Compliant categories without cloud dependency. A five-way classifier comparison on the same 80:20 stratified split (n = 100 records) shows that Random Forest (n = 100 trees) attains the best overall accuracy of 85.0% and macro F1-score of 0.681 among all methods tested. Its memory footprint, however, disqualifies it from the current ESP32 deployment; Naive Bayes, requiring only 96 bytes of parameter storage, achieves 80.0% accuracy and a macro F1 of 0.634 within the hardware's SRAM constraint. Per-class evaluation exposes a pronounced gap between the majority Compliant class (F1 = 0.903) and the minority Late and Non-Compliant classes (F1 = 0.500 each), a direct consequence of the 80:12:8 training distribution. All results are presented at prototype stage; generalisation to clinical populations requires prospective data collection under an institutional ethics protocol. Recommended priorities for future work are: (1) collecting at least 30 records per compliance class under ethics approval to close the minority-class deficit; (2) evaluating SMOTE and cost-sensitive training to narrow the Compliant–minority F1 gap; (3) porting a quantised integer-arithmetic Random Forest to the ESP32 once a balanced dataset of sufficient size is available; (4) integrating a Bluetooth-connected mobile application for caregiver dashboards and remote dose-schedule adjustment; and (5) implementing ESP32 deep-sleep scheduling between dose events to extend battery life beyond the current 6–8-hour active-use estimate.

References

[1] S.-J. F. Stewart, Z. Moon, and R. Horne, "Medication nonadherence: health impact, prevalence, correlates and interventions," *Psychol. Health*, vol. 38, no. 6, pp. 726–765, 2023, doi: 10.1080/08870446.2022.2144923.

[2] Y. Gu *et al.*, "Predicting medication adherence using ensemble learning and deep learning models with large scale healthcare data," *Sci. Rep.*, vol. 11, p. 18961, 2021, doi: 10.1038/s41598-021-98387-w.

[3] M. Mason, Y. Cho, J. Rayo, Y. Gong, M. Harris, and Y. Jiang, "Technologies for medication adherence monitoring and technology assessment criteria: Narrative review," *JMIR mHealth uHealth*, vol. 10, no. 3, p. e35157, 2022, doi: 10.2196/35157.

[4] V. Peddisetti, P. K. Kandregula, J. A. John, S. Poomdla, K. George, and A. Panangadan, "Smart medication management: Enhancing medication adherence with an IoT-based pill dispenser and smart cup BT - 2024 IEEE First International Conference on Artificial Intelligence for Medicine, Health and Care (AIMHC)." pp. 137–144, 2024. doi: 10.1109/AIMHC59811.2024.00030.

[5] T. Islam, R. Hassan, S. R. Romy, D. Dellal, and T. R. A. Bin, "Enhancing medication adherence with IoT technology," *Eur. J. Electr. Eng. Comput. Sci.*, vol. 7, no. 5, pp. 7–13, 2023, doi: 10.24018/ejece.2023.7.5.557.

[6] P. Dayananda and A. G. Upadhya, "Development of Smart Pill Expert System Based on IoT," *J. Inst. Eng. Ser. B*, vol. 105, no. 3, pp. 457–467, 2024, doi: 10.1007/s40031-023-00956-2.

[7] R. T. Meghla, M. E. Deowan, A. K. Nuhel, M. M. Sazid, M. N. Ekbal, and M. H. Mahamud, "An Internet of Things (IoT)-based smart automatic medication dispenser with an integrated web application for patient diagnosis," in *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, 2022, pp. 16–21. doi: 10.1109/IC2IE56416.2022.9970073.

[8] N. R. Kadam, C. Pardeshi, D. Rokade, O. Sawant, and S. Udhane, "IoT based automatic medicine dispenser with alert system," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 10, no. 3, pp. 343–348, 2024, doi: 10.32628/IJSRCSEIT.

[9] A. A. Jillepalli, S. Vattigunta, J. Varma, and B. J. De La Cruz, "A portable voice assisted smart drug delivery system with automated pill dispensing for dementia patients," *Healthc. Technol. Lett.*, vol. 12, no. 1, 2025, doi: 10.1049/htl2.12100.

[10] Z. Zhao, A. Gkikas, E. Yfantidou, E. Chatzidimitriou, N. Vretos, and P. Daras, "Machine learning-

based classification of medication adherence among patients with noncommunicable diseases," *Digit. Heal.*, vol. 11, 2025, doi: 10.1177/20552076241313073.

[11] V. Govindaraj, V. Dinesh, R. Gowsick, E. Saravanan, and S. Y. Yugesh, "DOSEGUARD: An IoT-enabled smart drug dispenser for improving medication adherence and real-time monitoring," *Int. J. Eng. Res. Sustain. Technol.*, vol. 3, no. 2, pp. 9–16, 2025, doi: 10.63458/ijerst.v3i2.110.

[12] Tedyyana, A., and O. Ghazali. "W. Purbo, O.(2024). Enhancing intrusion detection system using rectified linear unit function in pigeon inspired optimization algorithm." *IAES International Journal of Artificial Intelligence (IJ-AI)* 13.2: 1526-1534.

[13] J. Wendl, A. Simon, M. Kistler, J. Hapfelmeier, A. Schneider, and A. Hapfelmeier, "Identification of target groups and individuals for adherence interventions using tree-based prediction models," *Front. Pharmacol.*, vol. 13, p. 1001038, 2022, doi: 10.3389/fphar.2022.1001038.

[14] E. A. Kassaw, A. K. Sendekie, B. M. Enyew, and B. B. Abate, "Machine learning applications to classify and monitor medication adherence in patients with type 2 diabetes in Ethiopia," *Front. Endocrinol. (Lausanne).*, vol. 16, p. 1486350, 2025, doi: 10.3389/fendo.2025.1486350.

[15] M. T. Ghozali, "Implementation of the IoT-based technology on patient medication adherence: A comprehensive bibliometric and systematic review," *J. Inf. Commun. Technol.*, vol. 22, no. 4, pp. 503–544, 2023, doi: 10.32890/jict2023.22.4.1.

[16] M. Li *et al.*, "Development and assessment of novel machine learning models to predict medication non-adherence risks in type 2 diabetics," *Front. Public Heal.*, vol. 10, p. 1000622, 2022, doi: 10.3389/fpubh.2022.1000622.

[17] D. Karagiannis, K. Mitsis, and K. S. Nikita, "Development of a low-power IoMT portable pillbox for medication adherence improvement and remote treatment adjustment," *Sensors*, vol. 22, no. 15, p. 5818, 2022, doi: 10.3390/s22155818.