International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

AI-Enabled IoT Health Monitoring with Local Storage and Remote Caregiver Alerts

Niranjan N S[1], Soumyadeep Das[1], Tejas Y[1]*, Dr Venkatesha G[2]

[1] Department of CSE (Artificial Intelligence and Machine Learning)
[2] Professor and Head, Department of CSE (AI&ML) and CSE (IoT, CyS including BCT)
East West Institute of Technology, Bengaluru, India

*Corresponding author: tejasy1ewit@gmail.com

Abstract - This paper presents an academic health-monitoring prototype that combines sensor acquisition, local storage, preliminary classification and caregiver email notifications. A Raspberry Pi collects heart-rate and oxygen-saturation inputs from a MAX30100 and temperature from a DS18B20. Analog pulse and ECG interfaces are also included, although measured ECG is not used for classification. SQLite and Flask support local records and a web dashboard, while ThingSpeak supplies an Android interface with vital cards, chatbot and healthcare-locator screens. A Random Forest assigns seven synthetic-data labels, and Gemini provides explanatory text. The documented experiment reports 95.70% accuracy on 2,000 synthetic test records using a 100-tree classifier. This result measures agreement with synthetic labels, not clinical diagnostic accuracy. Source inspection also identifies replacement readings, calculated blood pressure and demonstration presets that limit interpretation of displayed values. The contribution is an integrated engineering workflow and a clearly bounded software evaluation. Controlled comparison with reference instruments and patient-level validation remain necessary before clinical use.

Key Words: IoT health monitoring, Raspberry Pi, Random Forest, local storage, Android application, caregiver alerts.

1. INTRODUCTION

When a person needs repeated observation, readings from several devices must be collected, reviewed and communicated. A caregiver may be away from the bedside, while an internet interruption may interrupt remote viewing. A useful monitoring workflow should therefore retain local records and make the source and age of each reading clear. It should also distinguish an abnormal measurement from a failed sensor.

The engineering goal is to connect sensor inputs, local records, preliminary assessment and remote viewing in one prototype. The web dashboard presents heart rate, oxygen saturation, temperature, ECG status and calculated blood-pressure fields. Email communicates threshold warnings and assessment results. The Android app adds chatbot and facility-locator interfaces.

The contribution is the integration and evaluation of these components. It is not a new diagnostic algorithm or a claim of superiority over commercial monitors. The paper separates sensor processing, demonstration behaviour and synthetic-data performance so that each result can be interpreted within its actual scope.

1.1 Design objectives

The first objective is to bring the available vital fields into a common record that can be stored and displayed locally. The second is to make those records accessible through a web dashboard and an Android interface. The third is to connect a request for preliminary assessment to feature preparation, classification and an explanation. The fourth is to provide caregiver email output while keeping threshold checks separate from the generated explanation. These objectives describe the intended engineering functions; their implementation does not by itself establish that the measurements are clinically reliable.

The scope is a laboratory prototype using development sensors and a synthetic training dataset. Local record access is useful when remote services are unavailable, but offline storage alone does not guarantee uninterrupted acquisition or eventual cloud delivery. The present evaluation therefore treats local persistence, cloud communication, model classification and interface demonstration as separate parts of the system.

2. RELATED WORK

Yadav et al. investigated continuous contactless vital-sign monitoring in a hospital setting [1]. That study supports the need to evaluate monitoring against patient-level evidence, but its results cannot validate the different hardware used here. Random Forest is an established ensemble method for learning relationships in tabular data [2]. Evaluation must also prevent information from the test set entering preprocessing or model selection [3]. These considerations guide the distinction between our synthetic-data experiment and future clinical testing.

Existing monitoring studies establish the importance of reliable measurements and timely communication. This project explores a smaller engineering question: how local storage, mobile viewing and an explainable assessment interface can be connected using development hardware. Its evaluation does not reproduce a hospital trial or establish a clinical advantage over wearable devices.

3. SYSTEM ARCHITECTURE

Figure 1 separates acquisition from on-demand assessment. The collection process prepares telemetry, stores records, checks thresholds and uploads data. A prediction request submits vital fields to Flask, which prepares model features, obtains a class label and requests explanatory text. Threshold email does not depend on the language model.

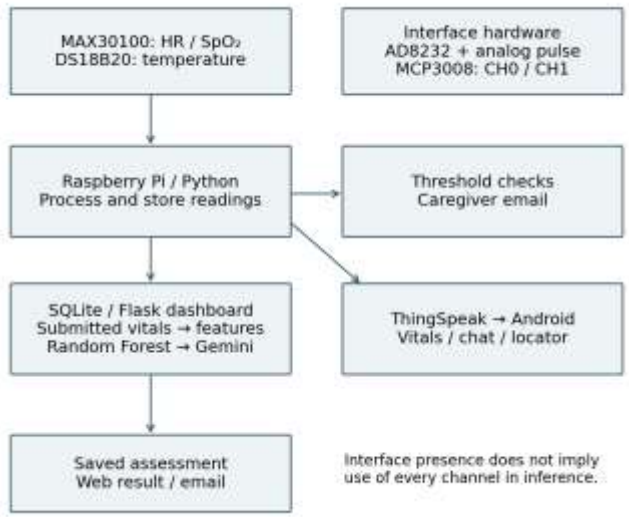

Fig -1: Acquisition, application and assessment paths.

3.1 Hardware and local storage

Table 1 lists the sensing and processing modules. MAX30100 provides optical pulse signals [4]; DS18B20 is a digital thermometer [5], not a thermistor. AD8232 is an analog ECG front end [6]. The additional analog pulse module connects to MCP3008 channel 1, while the ECG interface uses channel 0. The examined main collection loop obtains heart rate from MAX30100; two sensors do not imply automatic fusion or independent clinical verification.

Table -1: Hardware and interfaces

Component	Interface	Current role
Raspberry Pi 4 Model B	Linux, GPIO	Runs Python acquisition, Flask and SQLite.
MAX30100	I²C bus 1; address 0x57	Red and infrared samples for heart rate and SpO_2.
DS18B20	1-Wire	Digital temperature with communication CRC checks.
AD8232	Analog output; LO+ / LO−	Single-lead ECG front end; lead-off GPIO inputs BCM 17 and 27.
Analog pulse sensor	MCP3008 CH1	Separate pulse interface; not the main loop HR source.
MCP3008	SPI; CH0 / CH1	10-bit conversion for ECG and analog pulse interfaces.

The daemon writes both instance/health.db and sensor.db. The Flask application first attempts to read recent telemetry from sensor.db and then uses the main database as a fallback. SQLite provides local record access without a cloud database. Record timestamps and field order must remain consistent across both stores.

3.2 Cloud and mobile access

ThingSpeak fields 1–6 carry temperature, systolic pressure, SpO_2, heart rate, ECG category and diastolic pressure, respectively. The reviewed code makes a blocking HTTP request with a 10-second timeout and then sleeps for 15 seconds. Acquisition and network time therefore add to the interval. MathWorks documents a minimum 15-second update interval for free channels [7]; this prototype does not guarantee an exact 15-second end-to-end refresh.

Local database review can operate without a cloud connection. Remote telemetry, new Gemini responses and network notifications need connectivity. The latest Android interface shows a five-second refresh setting; reading the channel more often does not increase the sensor upload rate. Offline chatbot behaviour and complete outage recovery have not been tested.

3.3 Methodology and execution sequence

The methodology follows two connected sequences. In the acquisition sequence, Python requests values from the available sensor drivers, applies the configured processing, prepares a record and writes it to local storage. The collection process also evaluates threshold conditions and sends telemetry to ThingSpeak. In the assessment sequence, Flask receives submitted vital fields, constructs the expected feature vector and calls the classifier. The label and score are then passed with the vital fields to the text-generation service, and the resulting assessment is presented and saved.

This separation matters because the two sequences can operate at different times. A threshold check acts on a processed collection record, whereas an assessment acts on the values submitted in its request. An assessment should therefore not be assumed to describe the newest sensor sample unless its input timestamp is checked. The current screenshots do not establish that every interface shares a single synchronised record. Adding explicit record identifiers to the request, displayed result and email would make that relationship easier to verify.

The five-second Android refresh and the cloud upload interval refer to different operations. A refresh requests the latest available cloud entry; it does not force the Raspberry Pi to collect a new measurement. Several refreshes can therefore show the same record. A future interface should display both the measurement time and the last successful fetch time so that users can distinguish an unchanged reading from a communication delay.

4. SENSOR PROCESSING

The processing details below describe the examined source revision. They document how demonstration values were produced; they must not be interpreted as calibration or proof of accurate patient measurements.

4.1 Heart rate and oxygen saturation

The MAX30100 driver assumes 100 samples per second and collects approximately five seconds of optical data, with an eight-second buffer. It requires at least 300 samples and a mean infrared level above 5,000 over the latest 100 samples to proceed. These are software contact and sample checks, not calibrated measures of signal quality.

Heart rate is estimated from optical peak intervals after a five-sample moving average. The median accepted interval gives HR = 60 / interval in seconds. The collection process also applies an exponential moving average with weight 0.20 for the new value and 0.80 for the previous value. These filters reduce short-term variation but may delay a change; no reference-device experiment establishes motion-artifact rejection.

SpO_2 uses the original red and infrared arrays. For each channel, the mean is the DC component and the root-mean-square deviation is the AC component. The ratio R = (ACred/DCred)/(ACir/DCir) feeds the empirical estimate

SpO_2 = 110 − 25R. The driver limits this estimate to 50–100%. This relationship has not been calibrated against a reference oximeter for the assembled device.

The examined demonstration pipeline replaces some missing or out-of-range values and constrains displayed outputs. In particular, low oxygen inputs can be replaced by values in a normal-looking range. This is presentation behaviour, not noise correction, and can prevent a low-oxygen alert. Raw measurements, replacement values and simulations must therefore be distinguished in any future deployment.

4.2 Temperature

The DS18B20 routine reads the Linux 1-Wire device file, requires a successful CRC marker and converts the t= field from thousandths of a degree Celsius. It allows five attempts, with a 0.2-second delay after unsuccessful attempts, rounds to one decimal place and accepts 0–60 °C at driver level. The daemon replaces missing values or values outside 35–41 °C with 36.6–37.1 °C. CRC helps detect corrupted communication; it does not verify probe placement, body-temperature accuracy or agreement with a clinical thermometer.

4.3 ECG and calculated pressure

When hardware acquisition is enabled, the ECG routine requests a nominal one-second window at 50 samples per second and checks LO+ and LO−. The MCP3008 supplies counts from 0 to 1,023; these are not calibrated millivolts. Amplitude and variance rules assign categories 0–4, with 4 representing lead off. This heuristic is not a validated rhythm-analysis method.

Measured ECG is excluded from classification because its readings were unstable. The supplied runtime enables ECG simulation by default; the waveform endpoint also uses a simulator. The classifier receives the normal ECG category instead of a measured category. Consequently, this implementation does not demonstrate ECG-based arrhythmia detection or an end-to-end lead-off rejection mechanism.

No blood-pressure instrument is installed. Systolic and diastolic fields are calculated from processed heart rate with small random variation. They are demonstration values, not measured blood pressure. Pulse pressure and mean arterial pressure calculated from these fields inherit the same limitation.

Smoothing, communication checks, output limits and replacement values serve different purposes. Only the first two address fluctuation or transfer integrity directly. No paired reference readings, raw optical recording set or calibrated patient ECG dataset were available to calculate sensor accuracy for this prototype.

4.4 Interpretation of filtering and signal checks

The moving average operates on a short sequence of optical samples, while the exponential moving average operates on successive heart-rate estimates. Both reduce rapid variation, but neither can determine whether the underlying signal came from a correctly positioned sensor. A smooth output may still be wrong. Similarly, a successful temperature CRC verifies the integrity of the transferred bits, not whether the probe represents the intended measurement site.

The additional analog pulse interface offers a possible future comparison channel. Before combining its output with MAX30100, both signals would need matched timestamps and independent quality checks. Simply averaging their heart-rate estimates could hide a poor channel rather than improve the result. The present implementation does not claim this comparison or fusion. A suitable next experiment would record both raw signals under stationary and motion conditions and compare each estimate with a reference instrument.

Lead-off inputs provide connection information for the ECG interface. Connection status and useful ECG quality are separate issues: attached electrodes can still produce a noisy signal. Because measured ECG is excluded from classification in this version, the synthetic arrhythmia score cannot be used as evidence that the hardware identifies abnormal rhythms. The sensor interface, waveform display and model input must be evaluated independently.

5. CLASSIFICATION AND INTERFACE

The model schema contains six input fields: heart rate, systolic pressure, diastolic pressure, temperature, SpO_2 and ECG category. Six additional features are used by the 12-feature model, as listed in Table 2. These transformations express relationships already present in the inputs; they do not provide independent sensor evidence.

Table -2: Engineered model features

Feature	Definition
Pulse pressure	SBP − DBP
Approximate mean arterial pressure	DBP + (SBP − DBP)/3
Fever indicator	1 when temperature ≥ 38 °C; otherwise 0
Hypoxia indicator	1 when SpO_2 ≤ 90%; otherwise 0
Tachycardia indicator	1 when HR > 100 bpm; otherwise 0
Systolic-pressure stage	2 when SBP ≥ 140; 1 when 130 ≤ SBP < 140; otherwise 0

The seven training labels are healthy, chest_pain, fever, hypoxia, hypertension, arrhythmia and improper. They are synthetic-data classes, not seven verified diagnoses. Chest pain is not directly measured by these sensors. The displayed confidence is the highest model probability multiplied by 100; it is not measurement accuracy or a validated probability that a patient has a disease.

5.1 Generated explanation

The application sends the vital fields, predicted label and score to Gemini. Its configured primary model is Gemini 2.5 Flash, with Flash Lite as a fallback. The response is saved as explanatory text. The code does not implement a validated accept/reject decision or a clinical override of the Random Forest result. The API provides text-generation capabilities [8]; medical correctness and usefulness of the resulting explanations require separate assessment.

Figure 2 shows the supplied web dashboard. Table 3 records its visible fields and clarifies their meaning. A screenshot demonstrates interface presentation; it does not establish measurement accuracy.

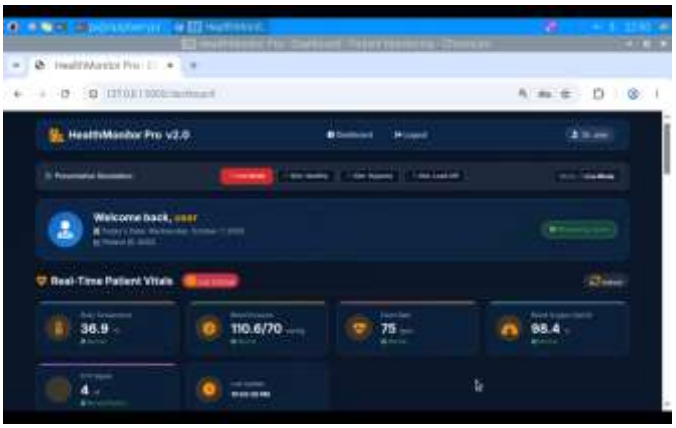

Fig -2: Supplied web dashboard demonstration.

Table -3: Fields shown in the dashboard snapshot

Field	Displayed value	Interpretation
Heart rate	75 bpm	Optical estimate after processing or replacement.
SpO_2	98.4%	Optical estimate after processing or replacement.
Temperature	36.9 °C	Probe value or fallback; the UI does not distinguish the source.
ECG	Category 4 / 4	UI says normal; driver category 4 means lead off.
Blood pressure	110.6 / 70 mmHg	Calculated systolic and diastolic presentation fields.

5.2 Android application

The Flask dashboard provides vital cards, trends, prediction history. The updated Android app connects a telemetry dashboard to an AI Medical Assistant chat screen and a Healthcare Locator & Booking screen (Fig. 3). Visible chat actions include My Health Status, Read Sensor Vitals and Diet & Medication, plus a text box and Send button. The locator presents facility filters, addresses, displayed distances, call controls and appointment or consultation buttons.

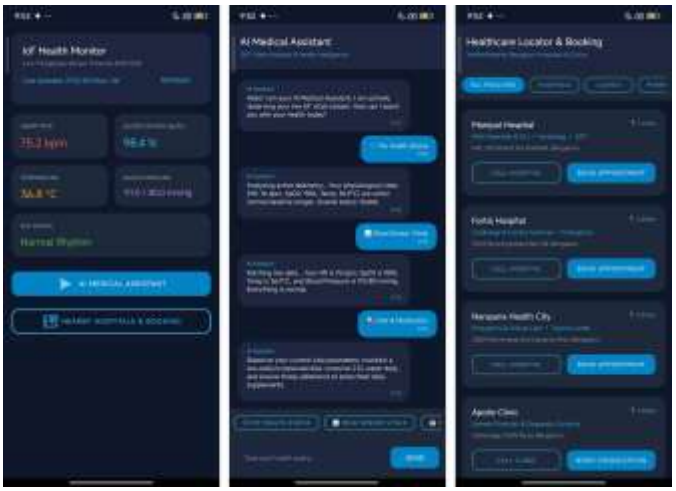

Fig -3: Android vital cards, chatbot and healthcare locator.

The Android manifest identifies MainActivity, ChatbotActivity and HealthcareActivity, with internet and network-state permissions. The supplied screens show chatbot responses, facility information and contact or booking controls. They do not verify completed appointments, location accuracy or consistent binding of every chat response to the latest record. These functions require end-to-end testing before claims of dependable service.

5.3 Email and demonstration modes

The current project uses caregiver email for threshold warnings and a separate application path for assessment results. Assessment emails communicate a preliminary model output and explanation for human review; they are not confirmed diagnoses. The examined threshold-email configuration uses a 30-minute cooldown to reduce repeated messages. Delivery success, delay and failure recovery require separate measurement.

Table -4: Examined threshold rules and processing limitations

Field	Condition in code	Effect of current processing
Heart rate	> 130 or < 60 bpm	High-HR condition can occur; the 65 bpm lower bound blocks low-HR alerts.
SpO_2	≤ 90%	Replacement and the 95.5% lower bound block this condition.
Temperature	≥ 38 °C	38–41 °C can trigger; readings above 41 °C are replaced.
Blood pressure	SBP ≥ 160 or DBP ≥ 105	Calculated output bounds prevent both conditions.
ECG	Category 3	Default simulation prevents this condition; no separate category-4 alert in this loop.

The conditions in Table 4 describe software rules, not patient-specific clinical recommendations. Notification code and a delivered email screenshot do not demonstrate reliable detection of a physiological emergency. Any deployment would first require quality-aware inputs, validated rules and measured delivery performance.

Preset modes provide selected values and prepared responses for repeatable interface demonstrations. In the examined revision, they bypass live classifier inference and the Gemini request. They should therefore be labelled as simulations and evaluated separately from dynamic sensor operation. A successful demonstration does not establish detection of a physiological emergency.

5.4 Handling assessment output and service failures

The Random Forest output is a class selected from its training labels. The generated explanation adds readable text, but it is not an independent measurement. Sending the same input fields to a second model does not create new evidence about the patient. An explanation can therefore repeat a limitation already present in the input, including calculated pressure or a substituted value. Human review remains necessary, and the result should retain the input values used for that particular assessment.

Threshold emails and assessment emails also have different meanings. A threshold email indicates that a configured rule was satisfied by a processed value. An assessment email reports the classifier output and accompanying explanation. Neither confirms that a recipient has read the message or taken action. The cooldown reduces repeated threshold messages, but its suitability for different situations has not been evaluated. Delivery acknowledgements, failed-send logging and escalation are possible improvements rather than demonstrated current functions.

The mobile chatbot and locator broaden the interface beyond vital cards. Their demonstrated contribution is access to conversation and facility-contact screens within the same application. The screenshots do not establish a verified medication service, a completed appointment transaction or a clinically reviewed response policy. Such claims would need backend source, execution logs and task-specific tests. Keeping these boundaries explicit avoids treating visible buttons as evidence of completed services.

6. EVALUATION AND RESULTS

6.1 Dataset and protocol

The supplied dataset contains 10,000 synthetic records, six input columns and one class label, with no missing cells or exact duplicate rows. Four classes have 1,429 records each and three have 1,428. This near balance is a dataset property, not an estimate of disease prevalence. Some input combinations are outside plausible physiological ranges, which further limits clinical interpretation.

The synthetic records contain generated vital fields and assigned class labels rather than observed patient diagnoses. The available materials do not fully document the original generation distributions or label-assignment rules. This limits reproducibility and means the model may largely learn relationships built into the generator. Publishing the generation script and an immutable dataset version is an important next step.

A fixed 100-tree Random Forest was reproduced using a stratified 80:20 split, random_state = 42 and n_jobs = 1. The 12 features were reconstructed and scaling was omitted for this tree-only experiment. No new parameter search was performed. The reproduction reused the development split, so it is not an external test. The original training workflow fits a scaler before splitting and compares model choices on test results; future evaluation should isolate training, validation and test data [3].

The reproduced model correctly classified 1,914 of 2,000 test records: accuracy 95.70%, macro F1 0.9570 and weighted F1 0.9571. Table 5 reports the class results, and Fig. 4 shows their F1 scores. The supplied SVM and KNN comparison values were not independently reproduced and are therefore excluded from the main result comparison.

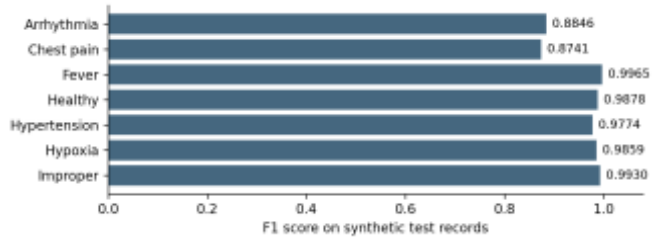

Fig -4: Class F1 scores on synthetic test records.

Table -5: Random Forest synthetic-data results

Class	Precision	Recall	F1	Records
arrhythmia	0.8815	0.8877	0.8846	285
chest_pain	0.8741	0.8741	0.8741	286
fever	0.9931	1.0000	0.9965	286
healthy	0.9861	0.9895	0.9878	286
hypertension	0.9690	0.9860	0.9774	285
hypoxia	0.9964	0.9755	0.9859	286
improper	1.0000	0.9860	0.9930	286

Arrhythmia and chest_pain have lower recall than the other classes. Their scores describe agreement with synthetic labels. They do not validate ECG analysis, establish clinical chest-pain detection or demonstrate the same performance when measured ECG is excluded from the application input.

6.2 Software checks

Processing checks were performed without hardware access or external services. With random seed 42 and a starting HR EMA of 75 bpm, missing HR, SpO_2 and temperature produced 75 bpm, 97.9% and 36.6 °C. An input of HR 75 bpm, SpO_2 85% and temperature 36.8 °C produced 75 bpm, 97.9% and 36.8 °C, confirming that the low oxygen input was replaced. HR 45 bpm with SpO_2 98% and temperature 39 °C produced 69 bpm, 98.1% and 39 °C.

A separate mock-classifier check confirmed that an ECG category of 4 was replaced by the normal category before classification. A five-sample moving-average check changed the sequence [10, 10, 10, 10, 60] to [10, 10, 10, 10, 20]. These tests verify specific arithmetic and routing behaviour; they do not measure sensor error, patient outcomes or end-to-end alert latency.

The evaluation covers synthetic-label classification and isolated software behaviour. It does not include patient recruitment, calibration against reference instruments or measurement of notification latency.

6.3 Meaning of the reported metrics

Accuracy is the number of correctly assigned synthetic labels divided by the total number of test records. Here, 1,914 divided by 2,000 gives 0.957, leaving 86 mismatched labels. Precision measures how many records assigned to a class actually carry that label in the dataset. Recall measures how many records carrying that label were recovered. F1 is the harmonic mean of precision and recall. These measures describe classification agreement and do not measure the error of a temperature probe or pulse sensor.

Macro F1 gives every class equal weight, while weighted F1 weights each class by its number of test records. Their similar reported values are consistent with the nearly balanced class supports in Table 5. Arrhythmia and chest_pain have lower F1 scores than the other labels. Without the complete confusion matrix, the results do not establish which alternative classes account for every error. Additional claims about the causes of those errors would therefore be speculative.

The reported experiment uses the complete model schema, including the ECG-category field. The application supplies a baseline category instead of measured ECG. Consequently, the held-out synthetic result is not an end-to-end performance

estimate for the deployed input path. A future comparison should evaluate the same classifier with the actual application inputs and, separately, a model trained only on dependable measured fields. The comparison must use an untouched test set and a predeclared evaluation procedure.

6.4 Validation needed beyond the present study

A practical validation plan should first test records and timing. A controlled sequence can check whether local storage preserves timestamps and field values, whether the dashboard retrieves the expected record, and whether the mobile app marks old data after a network interruption. These are proposed checks, not measurements completed in the present study. Reported outcomes should include the number of trials, failure conditions and observed delays rather than a single successful screenshot.

The next stage should assess sensor agreement using paired reference measurements and documented placement. Only after dependable acquisition is established should a patient-level classification study be considered. Its training and test groups should be separated by participant where repeated records are collected. Finally, generated explanations should be reviewed for consistency with the supplied fields and for unsupported advice. This staged approach keeps acquisition errors, classification errors and explanation errors distinguishable.

7. LIMITATIONS AND FUTURE WORK

The prototype combines local sensor interfaces and storage with an inspectable ML pipeline, generated explanations, Android vitals and chatbot screens, facility contact controls, and caregiver communication. This is an integration contribution. No direct benchmark establishes superiority over smartwatches or commercial monitors. Evidence remains limited to reviewed source behaviour, screenshots and a synthetic classification experiment.

The first improvement should preserve raw readings and attach source, timestamp and quality status to every value. A failed reading should be shown as unavailable, with a separate sensor warning, rather than replaced with a healthy-looking value. The ECG category labels must be made consistent across the driver, dashboard and presets. Calculated BP must be labelled as demonstration data or replaced by readings from a validated instrument.

Future work will replace the development sensors with suitable medical sensing devices and test them in a controlled environment. Paired measurements should compare HR and SpO_2 with a reference oximeter, temperature with a clinical thermometer, BP with a validated monitor and ECG with a reference recorder. The study should record measurement site, motion, signal loss and time alignment, and report error distributions, missing-data rates and agreement. Sensor accuracy and model performance must be evaluated separately.

Application tests should check stale mobile data, chat-to-record timestamps, unsupported responses, facility-list accuracy, call routing and booking outcomes, alongside network loss and message delays. Later ML evaluation needs appropriately labelled participant-level data and an untouched test set. Generated explanations and chatbot responses require factual and clinical review before use in patient decisions.

Electroencephalography (EEG) is a future research direction, not a current input. DEAP provides an example of physiological-signal research on self-reported affect [9], while Wu et al. investigated an EEG signature associated with antidepressant response in a specific clinical study [10]. Neither validates mood detection or treatment selection by this prototype. Future work could combine consented EEG recordings, patient reports and clinician review to study comfort or engagement during treatment.

8. CONCLUSION

The prototype connects sensor interfaces, local records, a web dashboard, remote Android viewing and caregiver email with a Random Forest assessment and generated explanation. The reported 95.70% accuracy applies only to synthetic-label classification. Replacement readings, simulated ECG and calculated blood pressure limit the present system to engineering demonstration and software study. Quality-aware acquisition, reproducible dataset generation and controlled validation are required before the workflow can support clinical decisions.

ACKNOWLEDGEMENT

The authors thank East West Institute of Technology, Bengaluru, and the departments of CSE (AI&ML) and CSE (IoT, CyS including BCT) for supporting this academic project.

REFERENCES

[1] A. Yadav et al., “Early detection of deteriorating patients in general wards through continuous contactless vital signs monitoring,” Frontiers in Medical Technology, vol. 6, Art. no. 1436034, 2024, doi: 10.3389/fmedt.2024.1436034.

[2] L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[3] Scikit-learn developers, “Common pitfalls and recommended practices.” [Online]. Available: https://scikit-learn.org/stable/common_pitfalls.html.

[4] Analog Devices, “MAX30100 pulse oximeter and heart-rate sensor.” [Online]. Available: https://www.analog.com/en/products/max30100.html.

[5] Analog Devices, “DS18B20 programmable resolution 1-Wire digital thermometer.” [Online]. Available: https://www.analog.com/en/products/ds18b20.html.

[6] Analog Devices, “AD8232 single-lead heart rate monitor front end.” [Online]. Available: https://www.analog.com/en/products/ad8232.html.

[7] MathWorks, “Channel data control.” [Online]. Available: https://www.mathworks.com/help/thingspeak/channel-control.html.

[8] Google, “Gemini API libraries.” [Online]. Available: https://ai.google.dev/gemini-api/docs/libraries.

[9] S. Koelstra et al., “DEAP: A database for emotion analysis using physiological signals,” IEEE Transactions on Affective Computing, vol. 3, no. 1, pp. 18–31, 2012, doi: 10.1109/T-AFFC.2011.15.

[10] W. Wu et al., “An electroencephalographic signature predicts antidepressant response in major depression,” Nature Biotechnology, vol. 38, pp. 439–447, 2020, doi: 10.1038/s41587-019-0397-3.