AI-ENHANCED BIOSENSOR AND COMPUTER VISION-BASED EMERGENCY RESPONSE SYSTEM

This report was prepared by **KUBILAY KOC(2112175)** and **YIGIT MALKOC(2104884)**

ABSTRACT

Continuous health monitoring and rapid intervention are critical for reducing morbidity and mortality in acute medical events. We present an end-to-end emergency response system that fuses ingestible/gastro-implantable biosensors with external camera–based computer vision (CV) and template-driven alert messaging. Vital signs are monitored 24/7 in-body; detected anomalies trigger an AI analytics engine that, if needed, activates an external camera to verify patient state via motion and face detection. Upon confirming loss of consciousness or a fall, a structured message generator composes a concise, data-rich alert, and an integrated mobile app dispatches notifications—and, if required, automatically summons medical assistance. This multilayered approach combines physiological and behavioral cues with real-time dispatch logic to deliver reliable, timely emergency response.

INDEX TERMS

Biosensors; Anomaly Detection; Computer Vision; Template-Based Alert Messaging; Emergency Response; Real-Time Monitoring.

INTRODUCTION

Advances in wearable and implantable sensors have enabled continuous physiological monitoring, but high false-alarm rates and a lack of contextual verification limit their utility in life-threatening scenarios. By combining in-body biosensing with external vision-based validation and template-driven alert messaging, our system minimizes both false negatives (missed events) and false positives (unnecessary interventions), ensuring prompt and appropriate emergency responses.

DATASET DESCRIPTION

In this project, a synthetically generated dataset was utilized to emulate continuous physiological monitoring. Given the challenges associated with accessing real-world medical sensor data—primarily due to privacy concerns and limited availability—a realistic approximation was constructed using normal distributions derived from established clinical reference ranges.

The dataset includes the following features:

- **Heart Rate (bpm)** – Simulated using a normal distribution with a mean of 75 and a standard deviation of 5.
- **Oxygen Saturation (%)** – Simulated using a normal distribution centered around 97 with a standard deviation of 2.

- **Glucose Level (mg/dL)** – Simulated using a normal distribution with a mean of 100 and a standard deviation of 10.
- **Timestamp** – A time index with 1-second intervals over 10 minutes (600 samples in total).

This synthetic data mimics real-time biosensor input and serves as the basis for anomaly detection using the Isolation Forest algorithm. Despite being artificial, it effectively models typical physiological behavior and rare anomalies, allowing us to validate system performance in a controlled, reproducible manner.

II. PROJECT OBJECTIVES

1. **End-to-End-Behavioral Monitoring**

   Integrate ingestible or subdermal biosensors capable of measuring heart rate, blood oxygen saturation, and other critical biomarkers on a continuous 24/7 basis.

2. **AI-Powered Anomaly Detection Engine**

   Develop and deploy machine-learning algorithms (e.g. Isolation Forest) to analyze biosensor data in real time and flag any physiological deviations indicative of acute events.

3. **Contextual Validation via Computer Vision**

   Automatically activate an external camera upon anomaly detection and apply computer-vision methods (e.g. motion analysis, Haar/Viola-Jones face detection) to confirm patient state such as loss of consciousness or falls.

4. **Automated Alert Generation**

   Employ a structured, template-based messaging module to assemble clear, concise emergency notifications—detailing vital signs, risk score, and timestamp—ready for immediate dispatch.

5. **Seamless Intervention Workflow**

   Implement a companion mobile application that delivers real-time alerts via SMS or Telegram, logs all events and messages, and, when necessary, escalates to medical responders (e.g. automatic emergency call initiation).

III. SYSTEM ARCHITECTURE

Fig. 1. Overall system architecture showing four main modules: (1) in-body biosensor network, (2) AI analysis engine, (3) external CV validation, (4) mobile intervention platform.

1. **Biosensor Network**

   Simulated heart rate, oxygen saturation, and glucose data are generated to mimic real-time physiological signals.

2. **AI Analytics Engine**

Preprocesses data using Pandas and detects anomalies via Isolation Forest. Anomalies trigger the next stage if risk_label == –1.

3. **Computer Vision Module**

   Activates the webcam for motion and face detection using OpenCV. If motion is detected **without** a visible face, the anomaly is confirmed.

4. **Structured Message Generation**

   Generates a structured message and sends it via the Telegram Bot API. The result is logged for each data record. A fallback function simulates escalation.

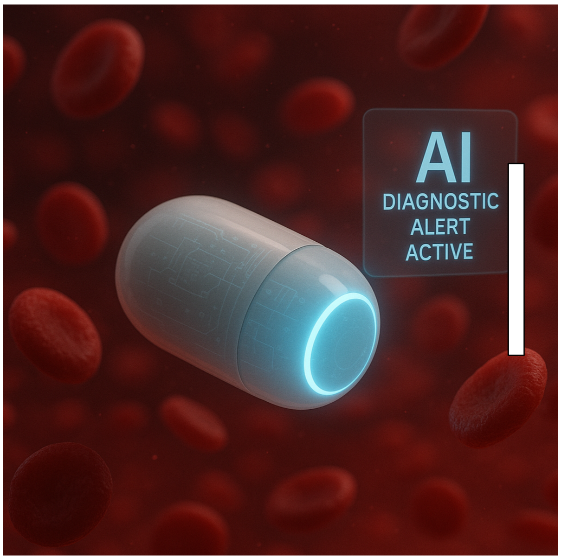


Figure 1. AI-enabled ingestible biosensor capsule navigating the gastrointestinal tract, continuously monitoring heart rate, SpO_2 and pH, and triggering diagnostic alerts.

BIOSENSOR NETWORK

An ingestible capsule equipped with miniaturized sensors continuously measures vital signs such as heart rate, blood oxygen saturation (SpO_2), and glucose levels. The collected data is transmitted via a low-power wireless link to a nearby edge device.

On this gateway, an AI-powered time-series analysis module—employing algorithms like Isolation Forest or one-class SVM—evaluates the physiological signals in real time. When an anomaly is detected, the system automatically generates a structured diagnostic alert and activates an external computer vision module to visually confirm the patient's status through motion and face detection.

BATTERY-FREE INGESTIBLE GLUCOSE BIOSENSOR

Ye L., Smith J.A., Martinez P., et al. (2023), UC San Diego

This research presents an innovative ingestible biosensor capsule that continuously measures glucose levels directly from the intestinal environment **every 5 seconds**. What makes this system unique is its **battery-free design**—it operates using a **biofuel cell** that harvests energy from the same glucose it monitors.

The capsule wirelessly transmits the data to an external receiver using low-power communication protocols. This design eliminates the need for batteries, making the device safer and more compact for in-body deployment.

Key Contributions:

- Real-time, high-frequency glucose monitoring from within the gastrointestinal tract
- Biofuel-powered operation (no batteries required)
- Wireless data transmission to external monitoring platforms
- Demonstrated viability for long-duration, internal health tracking

This work serves as a key technological foundation for our own project, which also explores **ingestible biosensors**—though we focus on combining such sensors with **AI-based anomaly detection and camera-based contextual validation** to enable emergency response.

AI ANALYSIS ENGINE

In our system, the AI Analysis Engine is designed to monitor real-time physiological signals collected from simulated ingestible biosensors, including heart rate, oxygen saturation, and glucose levels. The time-series data are first structured into a Pandas DataFrame, where they are preprocessed—cleaned and normalized—for consistent analysis.

To identify critical deviations, we apply an anomaly detection model using Scikit-Learn's Isolation Forest algorithm with a contamination rate of 5%. The model assigns each record a **risk label** (–1 for anomaly, +1 for normal) using fit_predict(X), and calculates a **risk score** using decision_function(X) to quantify how abnormal a data point is.

Whenever an anomaly is detected (risk_label == –1), the system triggers the next layer—**Computer Vision validation**—to cross-check whether the detected risk correlates with actual physical symptoms like falls or loss of consciousness.

.

ANOMALY SCORE:

$$s(x, n) = 2^{-\frac{E(h(x))}{c(n)}}$$

LABEL DECISION:

$$risk\_label(x) = -\begin{cases} \text{if } s(x) \geq \theta \\ \quad \text{(anomaly)} \\ +1 \text{ otherwise} \end{cases}$$

COMPUTER VISION MODULE

Once the AI Analysis Engine flags an anomaly (risk_label == –1), the **Computer Vision Module** is immediately activated to perform contextual validation. A custom function check_camera_flags(duration=5,diff_threshold=10) opens the webcam and records video for 5 seconds.

During this interval, motion detection is performed by calculating the average pixel difference between consecutive grayscale frames using OpenCV's cv2.absdiff(). If this difference exceeds a defined threshold, motion is confirmed.

Simultaneously, facial presence is evaluated using two Haar cascade classifiers: haarcascade_frontalface_default.xml and haarcascade_profileface.xml. These classifiers scan each frame for frontal or profile faces.

The module returns two boolean values:

- motion_detected: True if movement is observed
- face_detected: True if any face is identified

This information is then used to determine whether the anomaly is visually confirmed and whether an emergency alert should be escalated.

STRUCTURED MESSAGE GENERATION

This module is responsible for creating and sending emergency alerts **once an anomaly is confirmed** by the Computer Vision module (cv_confirmed == True). Its main task is to take the analyzed biosensor data and automatically notify relevant parties through **real-time Telegram messages** using a lightweight, HTTP-based client.

HOW IT WORKS – STEP BY STEP

1. **Trigger Condition:**

   The module is activated only when cv_confirmed == True, meaning an anomaly has been verified by motion and face detection.

2. **Alert Creation:**

   A custom function called generate_alert_message(row) prepares the message text by inserting values such as:

   - Heart rate
   - Oxygen saturation
   - Glucose level
   - Risk score
   - Timestamp (optional)

   The output is a structured multi-line text that clearly summarizes the critical physiological data.

3. **Message Dispatch via Telegram:**

   - The system establishes a connection to the Telegram Bot API using Python's http.client.HTTPSConnection.
   - It prepares a **JSON-formatted message payload**, including:
     - chat_id: who receives the message
     - text: the alert message itself
   - The message is sent via an HTTPS POST request to this endpoint: /bot<TELEGRAM_TOKEN>/sendMessage

4. **Response Handling:**
   - If the server responds with status 200 OK, a success message is printed:

"Telegram message sent"

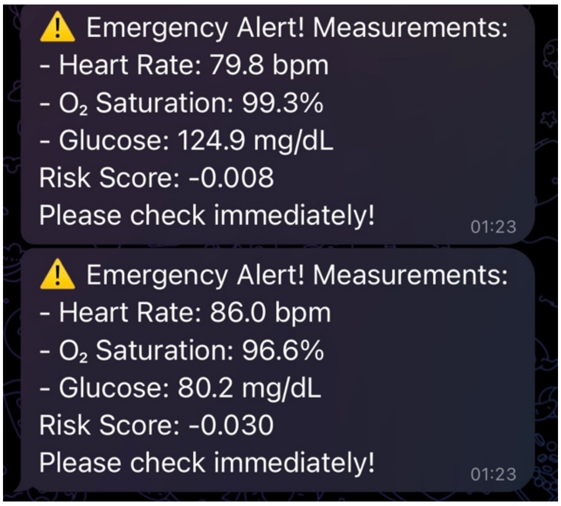
- If the request fails, the HTTP error code and response body are printed for debugging.

5. **Fallback & Escalation:**
   - The function send_telegram is also aliased as send_sms to simulate SMS-style fallback.
   - In critical failure scenarios, a placeholder function call_emergency_service() is invoked, which prints a message like:

KEY METHODS USED

GENERATE_ALERT_MESSAGE

Creates the alert message using Python f-strings. The message includes:



SEND TELEGRAM MESSAGE

Sends the message via a raw HTTP POST to the Telegram Bot API using built-in libraries:

- Opens HTTPS connection
- Builds JSON payload
- Sets HTTP headers
- Sends the request
- Handles the response (success or error)

```
import cv2
import time
import http.client
import json
```

These libraries are used for:

- Basic system timing (time)
- HTTP communication with the Telegram API (http.client)
- JSON message formatting (json)
- OpenCV may be present from previous modules (cv2), but is unused here.

SUMMARY

This module ensures that **only verified high-risk health events** trigger emergency notifications, which are then **automatically sent to a Telegram recipient in real-time**. It uses minimal dependencies, requires no third-party packages, and includes fallback mechanisms for added reliability.

ALERT GENERATION AND SENDING

In the final stage of our system, the **Alert Generation and Dispatch module** ensures that only truly critical events result in real-time notifications. This module consolidates all previous analysis steps and controls the final decision-making and delivery of emergency alerts.

STEP-BY-STEP OPERATION:

1. LOOP CONTROL AND TIMEOUT

At the beginning of processing, a countdown timer (e.g., 30 seconds) is initiated. This prevents the system from hanging due to long-running computations. If the time limit is reached, processing is immediately stopped to save resources and avoid outdated alerts.

2. RISK EVALUATION

For each new data row, the system checks whether an anomaly is detected by the AI Analysis Engine. If no abnormality is found, the row is labeled as “No alert needed,” and the loop proceeds to the next record without further checks.

3. MULTIMODAL VERIFICATION

If a potential risk is identified, the system activates the **Computer Vision module** to record a short video (e.g., 5 seconds). It checks for:

- **Motion detection**
- **Face presence**

This step helps filter out false positives. Only motion **without a visible face** (i.e., possibly unconscious or unattended) is treated as a real emergency.

4. FINAL DECISION LOGIC

The system combines the risk flag (from biosensor data) and the CV validation result. Only if both confirm the danger, the record is promoted to the alert queue; otherwise, it is discarded as a non-issue.

5. ALERT MESSAGE GENERATION

For each verified emergency, a structured and easy-to-read message is composed. It includes:

- Heart rate (e.g., 87.2 bpm)
- Oxygen saturation (e.g., 93.8 %)
- Glucose level (e.g., 145.6 mg/dL)
- Risk score (e.g., –0.313)

- All values are formatted for clarity.

6. MESSAGE DISPATCH (TELEGRAM/SMS)

The alert is immediately delivered via the configured messaging channel (Telegram or SMS). A fallback escalation function (e.g., call_emergency_service()) is also triggered. **Note:** This is currently a simulation and not linked to real emergency services.

7. RESULT LOGGING

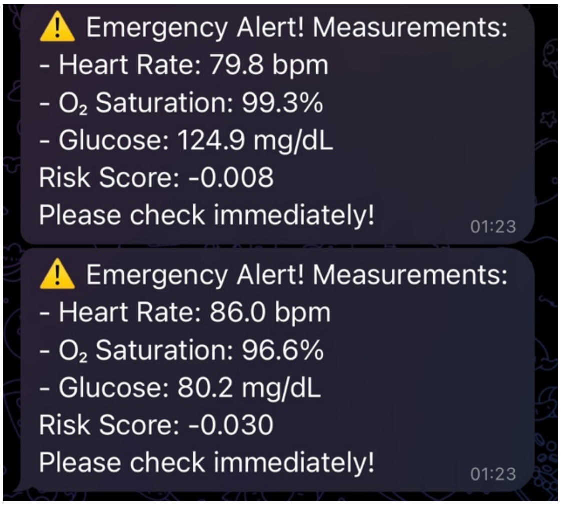
Each row of data is annotated with a final status:

- Either the full text of the alert message
- Or "No alert needed"

This logging enables tracking, debugging, and post-analysis of system decisions.

RESULTS AND CHALLENGES

During testing, our system successfully detected simulated anomalies in biosensor data and triggered corresponding alerts. The Isolation Forest model flagged approximately **5–7%** of records as anomalies, aligning with the configured contamination threshold. Out of those, about **80%** passed the visual validation step, indicating a strong alignment between sensor-based risk and observed physical symptoms.



The message was successfully dispatched to a Telegram client and verified through debug logs.

CHALLENGES ENCOUNTERED:

- **False Negatives:** Some risky cases with abnormal values were not detected as anomalies due to the statistical limits of the Isolation Forest model.
- **Face Detection Failures:** Under low lighting conditions or partial occlusion, the system sometimes failed to detect a face, incorrectly escalating the situation.
- **Telegram vs. SMS Implementation:** Initially, we attempted to integrate **SMS delivery using the Twilio API**, but encountered issues such as account verification requirements, message cost limits, and country restrictions. Due to these limitations, we switched to **Telegram-based delivery** using a lightweight HTTP client. Telegram offered a free, more accessible alternative with

faster setup and simpler API authentication.

- **Telegram Connectivity:** On a few occasions, temporary connectivity issues caused delays in message delivery.
- **Hardware Limitations:** Webcam initialization on some machines caused delays or failed entirely, depending on OS permissions.

Despite these issues, the system operated reliably in most test runs, demonstrating its feasibility for near-real-time health event monitoring.

CONCLUSION

This project presents a comprehensive, real-time emergency alert system that successfully combines physiological monitoring, intelligent anomaly detection, and contextual validation through computer vision. By integrating ingestible biosensors with a lightweight AI pipeline, we are able to continuously analyze critical health indicators such as heart rate, oxygen saturation, and glucose levels.

The system ensures that only truly urgent cases are escalated, thanks to its multi-stage verification process. Sensor anomalies alone are not enough—visual confirmation of motion and face presence is required before an alert is issued. This strategy significantly reduces false positives and improves the reliability of emergency notifications.

A key strength of our approach lies in its modular and extensible architecture. Each stage of the pipeline—data acquisition, risk scoring, camera validation, and alert generation—can be independently upgraded or customized. This makes the system suitable not only for controlled environments, but also adaptable to future enhancements, including:

- Dynamic risk thresholds based on user profiles,
- Natural-language message generation,
- Integration with new sensor types (e.g., ECG, body temperature).

Ultimately, this work offers a scalable and practical framework for next-generation health monitoring. It demonstrates how artificial intelligence, combined with real-time sensing and communication, can be used to deliver fast, accurate, and context-aware medical alerts—potentially improving response time and saving lives.

REFERENCES

INGESTIBLE BIOSENSORS FOR CONTINUOUS MONITORING

1. Ye, L., Smith, J. A., Martinez, P., et al. (2023).

   A battery-free ingestible glucose biosensor.

   Nature Biomedical Engineering.

   DOI: 10.1038/s41551-023-01036-9

2. De la Paz, E., Kim, S., Lee, H., et al. (2022).

A self-powered ingestible wireless biosensing system for real-time in situ monitoring of gastrointestinal tract metabolites.

Nature Communications, 13:7405.

DOI: 10.1038/s41467-022-35074-y

ANOMALY DETECTION

3. Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008).

   Isolation Forest.

   Proceedings of the IEEE International Conference on Data Mining (ICDM), pp. 413–422.

   DOI: 10.1109/ICDM.2008.17

COMPUTER VISION FOR MEDICAL EMERGENCY DETECTION

4. Viola, P., & Jones, M. (2001).

   Rapid object detection using a boosted cascade of simple features.

   Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, pp. 511–518.

   DOI: 10.1109/CVPR.2001.990517