Integrated Arduino and Machine Learning Patient Monitoring Platform

Presented by :
Ishita Mohanta; J. Tvisha; Tanisha Choudhuri

PROBLEM STATEMENT - Track 2

Develop a device for **continuous monitoring of at least two vital parameters** (such as body temperature, blood pressure, SpO_2, respiration rate, or heart rate) that can **provide real-time feedback and issue predictive alerts**. The system should enable early detection of abnormal conditions to support timely medical intervention.

Goal : Develop a low-cost, real-time system that measures vitals, predicts anomalies, and gives immediate feedback.

PROPOSED SOLUTION

Our model prototype helps us measure these key patient vitals :

1. **Heart Rate (in BPM)**
2. **Respiratory Rate (in Breaths PM)**
3. **Body Temperature (in Celsius)**

These **measured vitals are displayed on the LCD.**

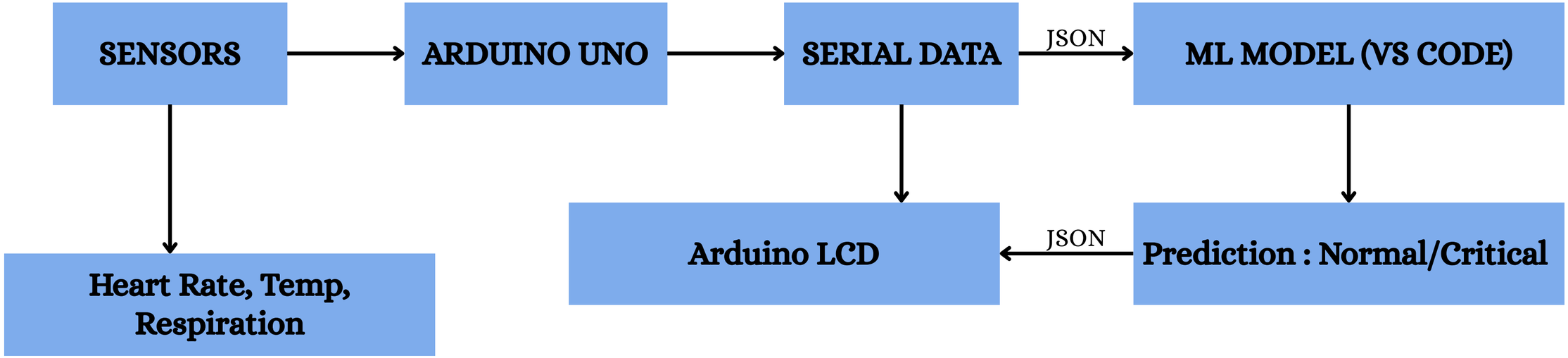
.

The data is processed in real-time using a trained **ML Model** which then helps us display the resulting alerts, hence providing actionable feedback for immediate intervention.

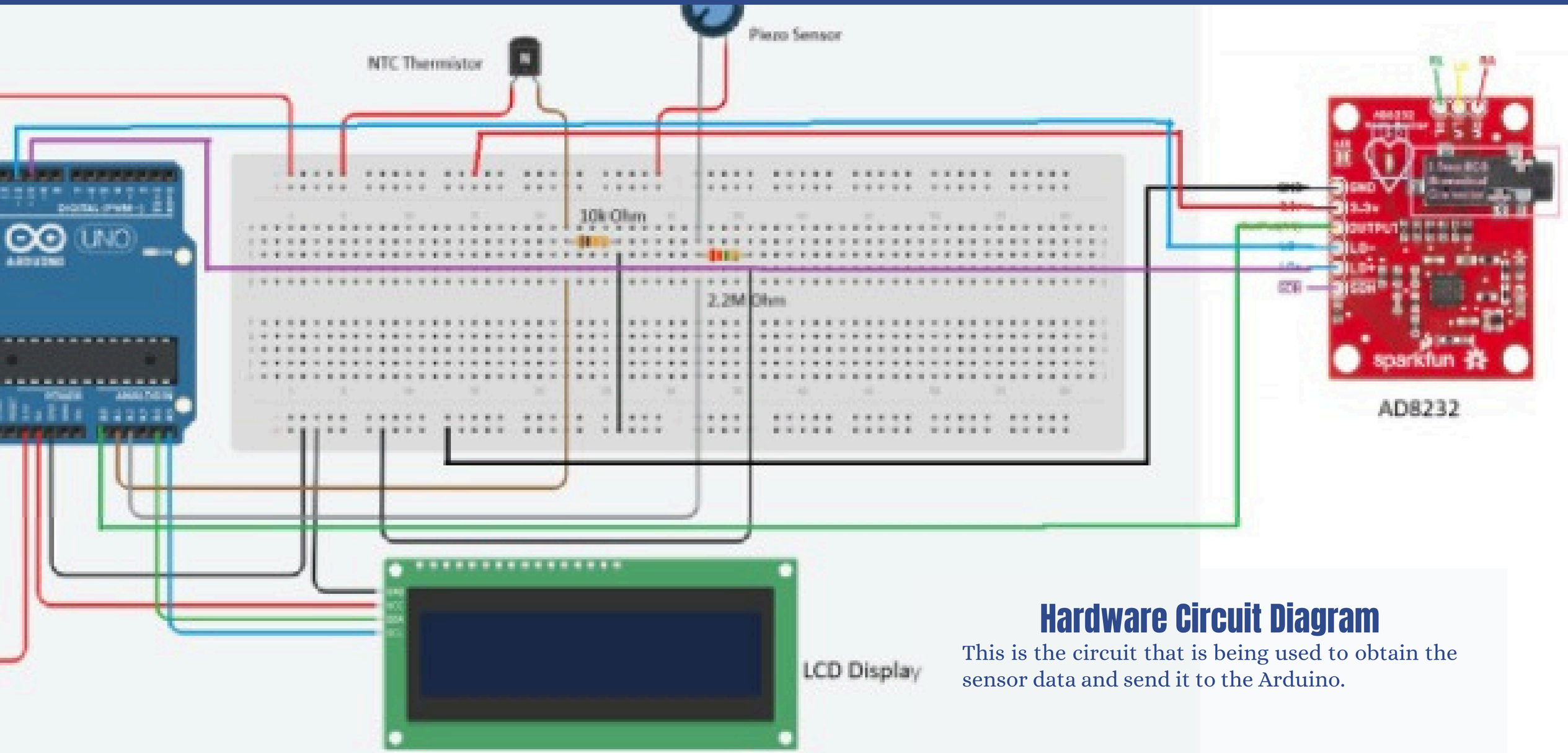
The **alerts are also displayed on the LCD** alternating at a regular interval.

SYSTEM OVERVIEW

https://github.com/TanishaChoudhuri/Integrated-Arduino-and-Machine-Learning-Patient-Monitoring-Platform



1. The sensors collect the analog data which the Arduino reads and converts into their respective physical units.
2. The data is sent via serial to the ML Model running on VS Code that predicts the patient status.
3. The feedback is sent back to the Arduino and the LCD displays the patient status which is helpful in real time monitoring along with the alerts alternating from time to time.



Hardware Circuit Diagram

This is the circuit that is being used to obtain the sensor data and send it to the Arduino.

HARDWARE COMPONENTS

01. ARDUINO UNO

Microcontroller used for data acquisition.

02. SENSORS

1. Heart rate sensor (**AD8232**)
2. **Piezoelectric** sensor for respiratory rate
3. **Thermistor** to measure body temperature

03. OUTPUT

I2C LCD 16x2 (displays the vitals as well as the alerts alternating from time to time.)

SOFTWARE AND ML INTEGRATION

01. C++ (ARDUINO IDE)

1. Reads the analog raw sensor values and converts them into real units and displays them on the LCD.
2. Helps alternate between the vitals display and the alert mechanism from time to time.

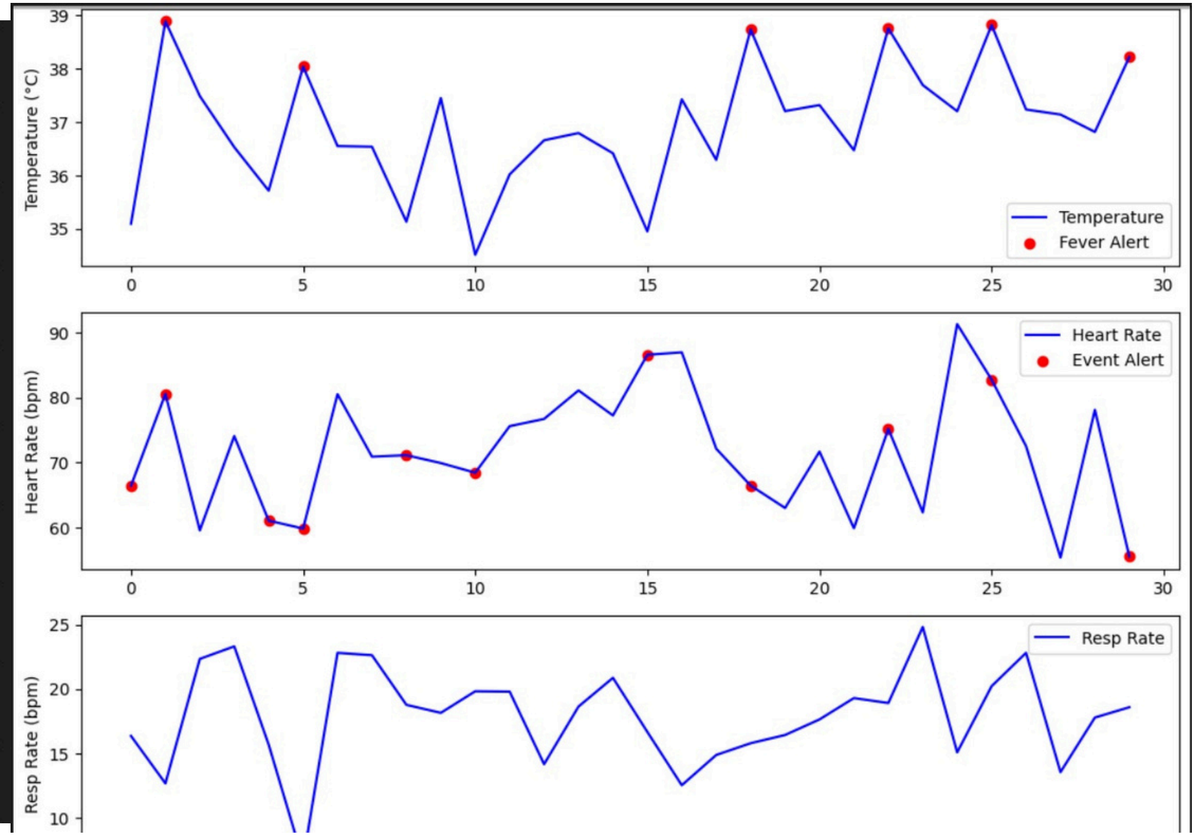
02. PYTHON (VS CODE)

1. Reads the Arduino data via serial port.
2. Uses a pre-trained ML Model to predict :
    a. Fever
    b. Event/Anomaly
    c. Respiratory Issues

TESTING AND RESULTS

1. Conducted tests on the hardware circuit using both simulated and real sensor data.
2. The machine learning model is trained using simulated data sets.
3. Heart rate, respiratory rate, and body temperature were measured accurately on the hardware circuit.
4. The vitals are displayed on the LCD for constant monitoring.
5. The hardware and the software were integrated by using serial communication between both the hardware and software.
6. The setup was tested multiple times in order to transfer the data correctly.
7. Machine learning predictions reliably identified expected anomalies.
8. The machine learning model also helps tabulate the data in real-time and also return feedback about the alert systems.

Temperature	Heart Rate	Resp Rate	Event	Fever	Respiratory
35.09	66.4	16.35	Alert	No Fever	Normal Resp
38.89	80.57	12.66	Alert	Fever	Normal Resp
37.49	59.53	22.33	Normal	No Fever	Normal Resp
36.53	74.06	23.3	Normal	No Fever	Normal Resp
35.71	61.03	15.67	Alert	No Fever	Normal Resp
38.04	59.81	6.67	Alert	Fever	Normal Resp
36.55	80.52	22.8	Normal	No Fever	Normal Resp
36.54	70.89	22.62	Normal	No Fever	Normal Resp
35.13	71.11	18.76	Alert	No Fever	Normal Resp

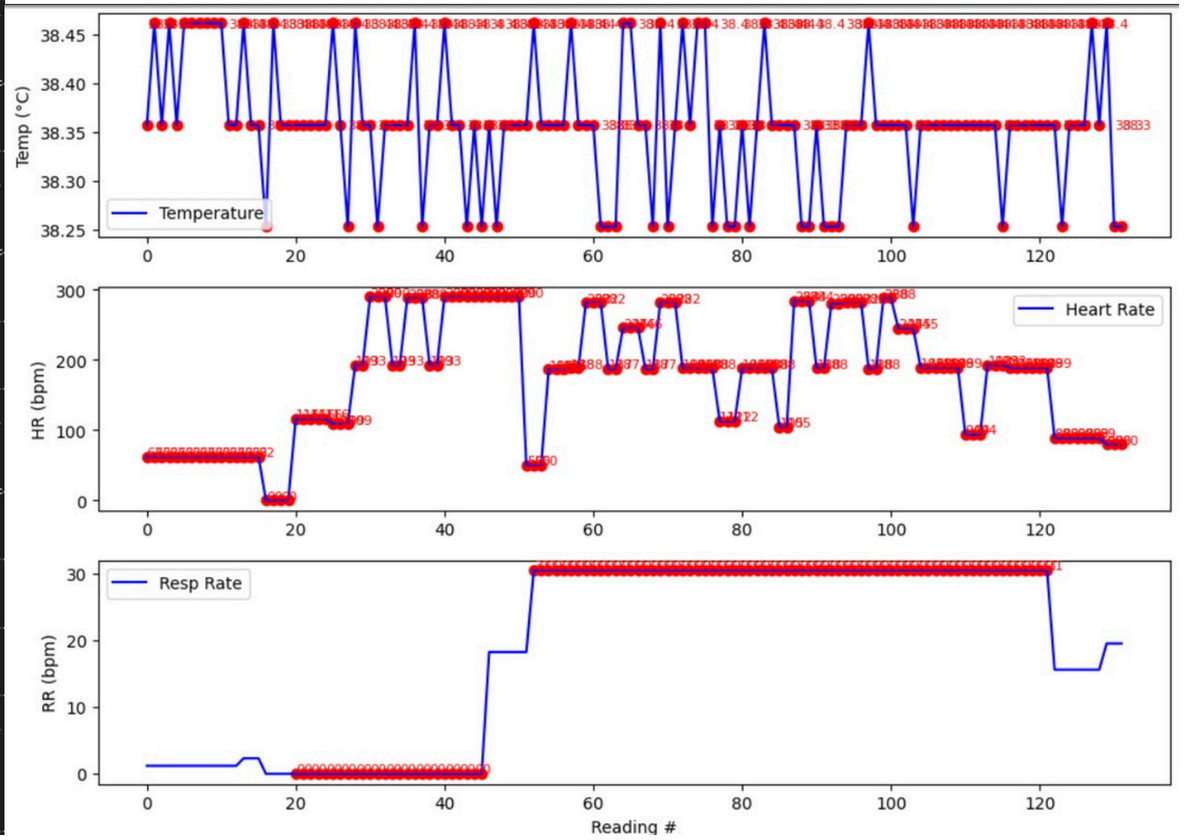


Results after using synthetic data

These data sets were used in order to train the ML Model before integrating the actual hardware circuit to it

The red nodes display the alerts while the blue lines display the real time data on the static graph.

Temperature	Heart Rate	Resp Rate	Event	Fever	Respiratory
38.4	62	1	Alert	Fever	Normal Resp
38.4	62	1	Alert	Fever	Normal Resp
38.5	62	1	Alert	Fever	Normal Resp
38.4	62	1	Alert	Fever	Normal Resp
38.5	62	1	Alert	Fever	Normal Resp
38.4	62	1	Alert	Fever	Normal Resp
...					
38.3	80	20	Alert	Fever	Normal Resp



Results after using live data

These data sets were obtained using the sensors and then processed by the trained ML Model in order to provide the real-time feedback.

The red nodes display the alerts while the blue lines display the real time data on the static graph.

FEATURES AND HIGHLIGHTS

Real-time measurement and prediction.

Predictive alert system for early intervention.

Low-cost, portable and easy-to-use

Works offline on a single laptop ; no internet is required for data transmission.

Data visualization (graphs) for patient trends.

FUTURE SCOPE AND IMPROVEMENTS

Incorporate ESP32 or other Wi-Fi modules to enable remote, real-time monitoring of patients..

Add additional sensors to measure SpO_2 (oxygen saturation) and blood pressure, providing a more comprehensive assessment of patient health.

Maintain a database of patient readings over time, enabling trend analysis, early detection of anomalies, and more accurate long-term health predictions.

Connect the monitoring system with hospital information systems (HIS) for automated data sharing, centralized record-keeping, and seamless clinical workflow.

CONCLUSION

Through this project, we developed a low-cost, portable real-time patient monitoring system with predictive ML alerts.

THANKS!

Do you have any questions?