RUTGERS HEALTH HACK 2024

Final Project Submission Instructions

Each project will be carefully evaluated based on the Judging Criteria outlined below. The submission is complete only when all elements in the checklist are submitted in the requested formats. Please follow the instructions carefully; projects with missing or partial elements will be automatically disqualified. After completing this document, do not forget to email it to **ruhealthhack@gmail.com** before 4 PM on Sunday, October 27, 2024. Any submissions received after the deadline will be automatically disqualified from judging.

Judging Criteria:

- **Innovation & Creativity:** How unique and original is the project?
- **Technical Implementation:** Quality of code, cloud architecture, and data integration.
- **Impact**: Relevance to healthcare and the potential real-world impact.
- **Presentation**: Clarity of the project demo and documentation.
- **Feasibility & Scalability:** Can this project be scaled up and deployed in a real healthcare setting?

Code Submission: All code must be uploaded and submitted via a version control platform (e.g., GitHub) by the deadline. Any updates or versions submitted after the deadline timestamp will be automatically disqualified.

Presentation/Demo: Teams must submit a brief project presentation or demo in video format (maximum of 5 minutes) explaining the project concept, technology stack, and healthcare impact. Upload the video to YouTube and ensure that appropriate permissions are set for access by the RUHealthHack team.

Working Prototype: A functional prototype/software or proof of concept is encouraged. It should demonstrate how cloud technologies and healthcare data are being utilized.

Documentation: Please complete the following document with all the requested information and send it to **ruhealthhack@gmail.com** by 4 PM on Sunday, October 27, 2024.

Team #: 19

Team Name: BIST

Team Lead Contact Information: Henry Shi, zhihengshi@outlook.com

Project Theme: AI Skin

Project Title: Alpha Skin

☐ Code Submitted

- Link to Github repository: https://github.com/huanyuchen0823/rutgers_hack_team19
- Overview of the solution:

Our innovative solution integrates advanced cloud computing with personalized wearable technology to monitor and correct spinal motion for individuals with Scoliosis and Kyphotic Deformities. By employing a robust Large Language Model (LLM) feature extractor in conjunction with piezoelectric textiles inspired by muscle fibers, we capture precise motion data that informs corrective feedback. The process begins with a cloud-based model that generalizes spinal motion patterns across users. This model serves as a shared intelligence resource, allowing it to continuously improve with new data while individual users benefit from personalized, locally trained models. The result is a federated learning system that adapts dynamically, providing both real-time corrections and optimized training routines for each user's unique needs.

Our system deploys motion detectors on critical spinal points—the shoulders, backbone, and cervical vertebrae—extracting high-resolution biomechanical data. This data is encoded and transmitted to the cloud, where it continuously refines the LLM feature extractor with collective insights from all users. Each personal device locally fine-tunes the model with its user's unique movement patterns, allowing the system to evaluate motion in real-time, determining if it aligns with corrective practices, and guiding users to achieve optimal form when deviations are detected.

- Details on cloud architecture and tools used

The cloud infrastructure serves as the backbone of our LLM feature extraction and motion analysis pipeline. Hosted on a robust cloud platform, our architecture provides a scalable, low-latency environment to ensure that sensor data processing and model updates are both fast and reliable. Key components of our cloud architecture include:

A. LLM Feature Extractor: The cloud-hosted Large Language Model (LLM) extracts critical features from motion data captured by the piezoelectric textiles. Designed for universal adaptability, this model learns from diverse data sources and continuously updates itself to improve accuracy across different motion patterns associated with scoliosis and kyphotic deformities.

B. Data Stream Processing and Storage: Data from sensors flows into the cloud via secure APIs, where it undergoes preprocessing and feature extraction. The data is stored in a highly scalable and secure database, enabling efficient retrieval and updating of the model with new datasets.

C. Model Synchronization Service: This module ensures the local personalized models receive the latest updates from the cloud model, keeping them in sync with new, generalizable motion patterns learned across different users. It also facilitates sending personalized model updates back to the cloud for reinforcement learning and continuous improvement of the general model.

Key Tools and Technologies:

A. Cloud Platform: AWS and Azure, providing scalability and integration with machine learning tools.

B. Machine Learning Frameworks: PyTorch or TensorFlow for model training and deployment.

C. Data Processing and Analysis: Apache Kafka for real-time streaming, coupled with data storage solutions such as Amazon S3 or Google BigQuery.

- Steps for installation or deployment

Step 1: Cloud Model Setup

Deploy the Cloud LLM Feature Extractor: Spin up a cloud environment with scalable GPU support and deploy the LLM feature extractor. Set up initial training data for the LLM, emphasizing motion features relevant to Scoliosis and Kyphotic Deformities.

Configure Containerization for Feature Updates: Establish a containerized environment for deploying model updates, ensuring the model can be continuously refined and pushed to local devices efficiently.

Step 2: Local Model Installation

Install Motion Data Detectors on the wearable piezoelectric textiles, positioning them on the shoulder, backbone, and cervical vertebrae.

☐ Set Up Local Processing Environment: Implement Edge AI protocols to handle real-time motion analysis locally. Each local device integrates the cloud-synced LLM model, which runs an evaluation loop to classify motion as "correct" or "incorrect." If a motion is classified as "incorrect," the model triggers corrective feedback, which is provided directly to the user. This feedback is generated by comparing the user's motion data to an idealized motion sequence and then guiding the user with precise adjustments (e.g., adjusting spinal alignment or shoulder positioning).

Step 3: Data Transmission and Model Update

A. Enable Secure Cloud Synchronization: Configure data pipelines using a reliable message broker for transmitting encoded motion data to the cloud, ensuring minimal latency. Implement secure channels with data encryption for safe data transfer.
B. Run Continuous Backpropagation: Each personal device transmits its refined motion data to the cloud, where updates to the general model occur using backpropagation. At regular intervals, this updated model syncs with personal devices to provide them with the latest feature extractions.
C. Launch Monitoring Dashboard: Deploy a real-time monitoring dashboard that visualizes user improvements and system updates, tracking each user's journey and enabling quick troubleshooting if needed.

In all, by converging the realms of wearable technology, cloud computing, and federated learning, this solution pushes the boundaries of personalized healthcare. It offers users a secure, continuously improving system that seamlessly bridges collective insights with individual precision, ensuring spinal health is always a step ahead.

Demo Presentation uploaded to YouTube with appropriate settings

- Link to YouTube Video: https://youtu.be/mW6-TsOQt3c