Personalized Fitness Recommendation and Tracking System

Dr. J. Angel Ida Chellam[1], Nambiaaruran S[2], Raghunath C G[3] and Sriraam K N[4].
[1] *Associate Professor, Sri Ramakrishna Engineering College, Coimbatore, India*
[2,3,4] *UG Scholar, Sri Ramakrishna Engineering College, Coimbatore, India*

Abstract--This **paper presents a comprehensive smart fitness system that merges embedded hardware and mobile software to deliver personalized workout planning and real-time strength training tracking. Utilizing the NodeMCU ESP8266 microcontroller and MPU6050 inertial measurement unit (IMU), the system captures and transmits motion data during barbell exercises. The Flutter-based mobile application employs machine learning models to classify exercises and accurately count repetitions and sets. Personalized workout and diet recommendations are generated based on user profiles including age, gender, medical conditions, and fitness goals. The system eliminates reliance on external servers by embedding models directly in the app, ensuring efficient and low-latency performance. Experimental evaluation indicates high accuracy in exercise classification and user satisfaction in recommendation quality.**

***Index Terms*—Fitness Tracking, Personalized Recommendation, NodeMCU, MPU6050, Machine Learning, Flutter, Barbell Exercises**

I. INTRODUCTION

The increasing popularity of digital health and fitness solutions has led to the development of various mobile applications and wearable devices. However, many existing systems provide generic fitness plans that do not consider the user's unique physiological attributes, medical history, or personal goals. Moreover, accurate tracking of barbell-based strength exercises remains a challenge due to limitations in conventional smartphone sensors and lack of real-time feedback mechanisms.

To address these gaps, we propose a smart fitness system that integrates a sensor-equipped wearable device with a mobile application to offer both personalized recommendations and accurate exercise tracking. The system leverages data from the MPU6050 sensor to monitor user motion during workouts and transmits it via the NodeMCU ESP8266 to the mobile device. Machine learning algorithms embedded within the app process this data to classify exercises and count repetitions and sets. Additionally, a recommendation module tailors workout and dietary plans based on individual user inputs.

This paper discusses the architecture, design methodology, implementation, and performance evaluation of the proposed system.

II. LITERATURE REVIEW

Stisen et al. (2015) explored how accelerometer data from wearable sensors can be used to recognize gym exercises, emphasizing the feasibility of activity recognition in real-life environments. Kassahun et al. (2019) proposed a framework for exercise recognition and repetition counting during free-weight workouts. Their system combined motion sensors with algorithmic processing, but lacked integration with personalized recommendations. Olivares et al. (2022) introduced a method that used chest-mounted accelerometers and LSTM neural networks to improve the accuracy of strength training recognition. Their focus was on deep learning techniques for classification but did not extend to end-user personalization. Smith et al. (2022) developed a machine learning-based system that utilized user health profiles to provide customized fitness recommendations. However, their approach did not integrate real-time sensor data for tracking physical activity. These studies highlight the progress and gaps in fitness tracking and personalization. Unlike previous efforts, our system merges both real-time sensor-driven tracking and personalized planning into a single, user-friendly mobile application.

III. PROPOSED SYSTEM AND METHODOLOGY

The proposed system is a hybrid solution that combines sensor-based motion tracking with machine learning-powered recommendations. It is composed of three core components: the hardware module, the Flutter mobile application, and the machine learning pipeline.

Hardware Module: The wearable hardware module uses a NodeMCU ESP8266 microcontroller connected to an MPU6050 sensor, which houses a 3-axis accelerometer and 3-axis gyroscope. This setup captures motion data during strength training exercises such as squats, deadlifts, and bench presses. Data is transmitted via Wi-Fi to the mobile application for further processing.

Data Acquisition and Preprocessing : Sensor data is continuously collected in real time from the MPU6050 sensor. To ensure the quality and usability of this data, several preprocessing steps are carried out. First, noise is filtered out using a Butterworth low-pass filter to smooth the signal. The raw data is then normalized to maintain consistency across input values. Statistical and frequency-based features, including mean, variance, skewness, kurtosis, Fast Fourier Transform (FFT), and signal energy, are extracted to capture the unique characteristics of each exercise. Outliers are identified and removed using the Interquartile Range (IQR) method and Local Outlier Factor (LOF) to preserve data integrity. Lastly, Principal Component Analysis (PCA) is applied to reduce dimensionality, making the dataset more manageable for machine learning model training while retaining essential information.

Mobile Application: The Flutter-based mobile application plays a central role in the system's operation by managing real-time communication with the hardware module through Bluetooth or Wi-Fi, ensuring seamless data transfer. It also visualizes the user's workout performance, displaying essential metrics such as exercise type, the number of repetitions, and sets performed. Additionally, the application provides an intuitive interface for users to input their health profile details, including age, gender, existing medical conditions, and specific fitness goals. This comprehensive functionality enables personalized tracking and recommendation features within the app.

Machine Learning Pipeline The machine learning component of the system consists of two parallel workflows: exercise recognition and personalized recommendation. For exercise recognition, models such as Random Forest, K-Nearest Neighbors (KNN), and Feedforward Neural Networks are trained using the preprocessed motion data collected from the MPU6050 sensor. The model that achieves the highest accuracy during evaluation is selected and embedded into the mobile application for real-time exercise classification.

For personalized recommendations, the system employs KNN and Cosine Similarity algorithms to generate suggestions for workout type, equipment, set and repetition count, and dietary plans. These recommendations are tailored to each user's health profile, fitness goals, and preferences.

A real-time feedback mechanism is integrated into the app, allowing users to rate the relevance and effectiveness of the suggestions. Ratings of 3.5 or higher are considered positive feedback and are used to fine-tune the recommendation model, enhancing its accuracy over time. Conversely, ratings below 3.5 signal dissatisfaction and are penalized during the model's retraining process to avoid repeating ineffective recommendations. This dual pipeline ensures that the system not only tracks performance accurately but also evolves to meet the individual needs of each user, creating a highly personalized fitness experience.

IV. IMPLEMENTATION AND RESULTS

The implementation phase began with collecting and preparing datasets from sensor readings generated during various barbell strength exercises, including squats, deadlifts, bent-over rows, shoulder presses, and bench presses. These raw sensor values underwent preprocessing, including filtering and transformation, to extract meaningful features. The final structured dataset was divided into training and testing subsets to train the classification and recommendation models.

Multiple machine learning algorithms were evaluated for performance, including Random Forest, K-Nearest Neighbors (KNN), Naïve Bayes, Decision Trees, and Feedforward Neural Networks. Grid Search was employed for hyperparameter tuning to improve model accuracy. Among these, Random Forest achieved the highest classification accuracy and was selected for deployment in the application. Model performance was assessed using standard evaluation metrics such as accuracy scores and confusion matrices, with results visualized through comparative bar plots.

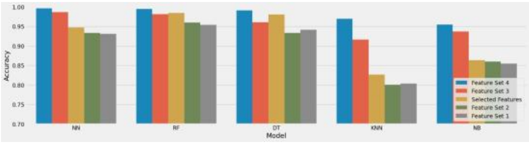

Figure 1: Accuracy comparison of different models using various feature sets.

The system demonstrated excellent classification accuracy in distinguishing between different exercises and showed strong consistency in counting sets and repetitions accurately.

On the recommendation front, the integration of feedback loops proved effective. The system dynamically adapted to user preferences based on ratings, improving the relevance of future suggestions. User testing confirmed high satisfaction levels in both workout tracking and personalized plan generation.

The model development pipeline is outlined in Figure 2.

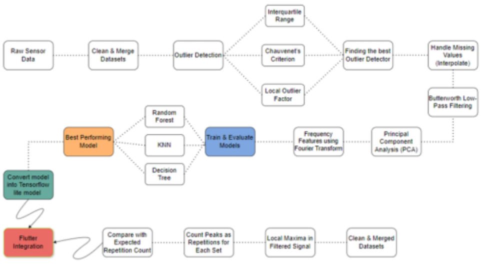

Figure 2: Machine learning pipeline and system integration.

Below are the visuals supporting the mobile app and hardware implementation:

Figure 3: Mobile app interface used to gather user profile information.

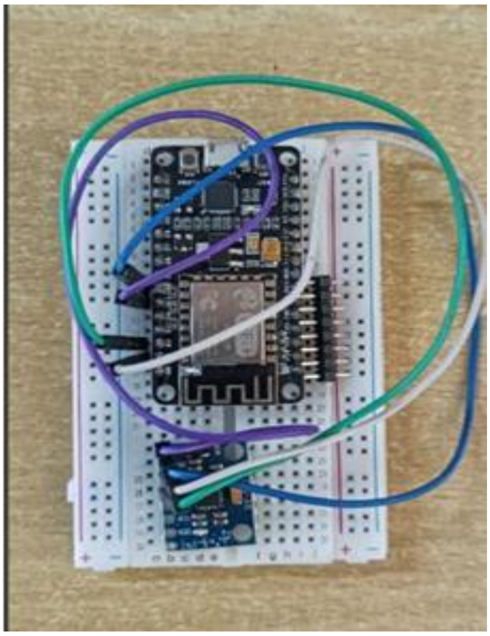

Figure 4: Hardware setup featuring NodeMCU ESP8266 and MPU6050.

Figure 5: System testing during a deadlift session.

The deployment of machine learning models directly within the Flutter application enabled real-time inference without the need for external servers, resulting in a smooth and responsive user experience.

V. CONCLUSION AND FUTURE WORK

The personalized fitness tracking and recommendation system presented in this study successfully integrates wearable hardware, mobile software, and machine learning to deliver a tailored and data-driven fitness experience. The real-time classification of barbell exercises and the ability to count repetitions and sets accurately make it a practical tool for strength training enthusiasts. The personalized recommendation module adds further value by aligning workout suggestions with the user's health profile and fitness goals.

Looking ahead, there are several promising avenues for future development. The system currently supports a limited set of barbell-based exercises; expanding its capabilities to include bodyweight, dumbbell, resistance band, and cardio exercises would broaden its utility. Moreover, incorporating voice-based AI coaching or large language models (LLMs) could enhance user engagement through intelligent interaction and adaptive support. Social features like fitness challenges, achievement sharing, and leaderboards could further improve motivation and user retention. With continued improvements, this system has the potential to evolve into a comprehensive digital fitness assistant tailored to individual needs.

REFERENCES

[1] Stisen, A., Blunck, H., Bhattacharya, S., Prentow, T. S., Kjærgaard, M. B., Dey, A., & Jensen, M. M. (2015). Recognizing Gym Exercises Using Acceleration Data from Wearable Sensors. *Proceedings of the 2015 ACM International Symposium on Wearable Computers*, 1–8.

[2] Kassahun, Y., Perrig, N., Daróczy, B., Widmer, S., Leutenegger, S., & Fua, P. (2019). Repetition Counting and Exercise Recognition in Free-Weight Workouts. *Sensors*, 19(714), 1–16.

[3] Olivares, R., Sundholm, M., Cheng, J., Lukowicz, P., & König, A. (2022). Exercise Recognition for Strength Training Using a Chest-Mounted Accelerometer and LSTM Neural Networks. *Sensors*, 22(7), 2489.

[4] Smith, J., Chen, L., & Patel, M. (2022). A Personalized Fitness Recommendation System Using Machine Learning and User Health Profiles. *IEEE International Conference on Healthcare Informatics (ICHI),* 1–8.

[5] Flutter – Build Apps for Any Screen. Available at: https://flutter.dev/

[6] Pub.dev – flutter_blue_plus (Bluetooth Plugin). Available at: https://pub.dev/packages/flutter_blue_plus

[7] Human Activity Recognition Using Wearable Sensors. Available at: https://www.sciencedirect.com/science/article/pii/S2352914820301032

[8] Google Fit APIs for Activity Tracking (conceptual reference). Available at: https://developers.google.com/fit

[9] UCI Machine Learning Repository – Activity Recognition Data. Available at: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones

[10] Healthline – Guide to Barbell Exercises and Proper Form. Available at: https://www.healthline.com/health/fitness/barbell-exercises