Rudramurthy V C, 2026, 14:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

FitSense: A Closed IoT Ecosystem for Gym Environments with RFID Tracking and ML-Driven Diet and Exercise Recommendations

Rudramurthy V C[1], J Madhav Pai[2], Gaurang Bharadwaj[3], Somya Das[4]
[1,2,3,4]Department of Computer Science and Business Systems, BMS College of Engineering, Karnataka, India

Abstract- Modern gym environments suffer from fragmented tracking ecosystems. Users typically rely on personal wearables to track workouts and separate mobile applications to log diets, while gym owners lack granular data on machine utilization. This paper introduces FitSense, a holistic closed IoT ecosystem designed to bridge this gap. By retrofitting standard gym machines with an ESP32-based IoT module featuring an RFID-RC522 reader and an MPU-9250/6500 inertial measurement unit (IMU), we enable seamless session tracking. Users tap their RFID cards to start and stop workouts on any machine. Instead ofcomplex motor instrumentation, the IMU calculates steps and speed based on step vibrations and the user's pre-configured stride length. This hardware ecosystem is tightly integrated with a centralized dietary and calorie tracker. A Machine Learning engine utilizes the combined history ofphysical exertion and nutritional intake to recommend personalized diet adjustments and future exercise routines. This creates a unified platform where fitness and nutrition are tracked, analyzed, and optimized in a single ecosystem.

Keywords- Internet of Things, Smart Gym, RFID, MPU-9250, Machine Learning, Diet Recommendation, ESP32

I. INTRODUCTION

In contemporary fitness tracking, data is largely siloed. The manager observes equipment usage empirically, while the member relies on personal wearables and self-reported diet logs [1]. None of this disparate data is intrinsically linked to the specific gym equipment being utilized. Consequently, gym owners struggle to monitor equipment health and utilization accurately, and members often lack a cohesive strategy that marries their actual physical exertion with their nutritional intake.

The first challenge is data isolation. A user might log calories in one application and track their heart rate on a smartwatch, but the gym infrastructure itself remains "dumb." The second challenge is the complexity of personalization. Fitness progress depends heavily on the synergy between the load applied during a workout and the dietary fuel consumed [2]. Existing gym management software rarely looks at both variables simultaneously.

FitSense tackles this by creating a closed ecosystem within the gym. Every machine is retrofitted with an inexpensive, standardized IoT module. When a member taps their RFID card, their session begins. The module captures step counts and inferred speeds using machine vibration, completely decoupling the sensor from the treadmill's complex internal motor electronics. This data is merged with the user's logged nutrition in a centralized application, allowing a Machine Learning (ML) model to suggest not only the next workout but also the dietary intake required to support it.

II. RELATED WORKS

1. Wearables vs. Fixed Infrastructure

Motion sensors are ubiquitous in personal fitness, primarily housed in wristbands and smartphones. The MPU-6050 and MPU-9250 are common low-cost choices for 6-axis and 9-axis tracking [3]. However, wearables suffer from battery limitations and user non-compliance. Shifting the sensor to the machine frame transfers the tracking burden to the infrastructure, ensuring 100% compliance as long as the user taps in.

© 2026 Rudramurthy V C. his is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly credited.

2. Nutrition and Fitness Integration

Most fitness recommendation systems operate in isolation. Apps recommend workouts based on self-reported readiness, ignoring nutritional deficits. Energy needs and recovery are mathematically linked to daily caloric intake [2]. FitSense introduces a paradigm where dietary adherence is a direct feature input for workout generation.

III. SYSTEM ARCHITECTURE

The FitSense architecture is divided into three primary nodes: the IoT Hardware Node, the Central Database & Application Layer, and the Machine Learning Engine.

1. IoT Hardware Node

To ensure cost-effectiveness and ease of deployment across various machine types, we shifted from a complex multi-sensor array to a streamlined, three-component architecture, illustrated in Figure 1.

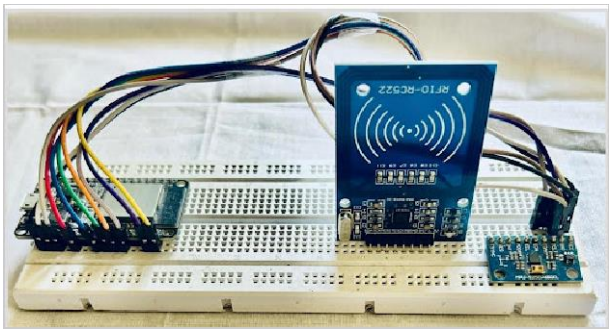


Fig. 1. The FitSense IoT prototype featuring an ESP32 microcontroller, RC522 RFID reader, and MPU-9250/6500 IMU on a breadboard.

The core controller is an ESP32, which provides native Wi-Fi capabilities, eliminating the need for a secondary gateway device like a Raspberry Pi for data transmission. Connected to the ESP32 via the SPI bus is an RFID-RC522 reader. This module scans MIFARE cards [4], linking the physical machine to the user's database profile.

The sensing element is an MPU-9250/6500 module connected via I2C. Instead of reading the treadmill belt directly, this sensor is mounted to the rigid frame of the equipment. It captures high-frequency vibration data generated by the user's footfalls.

2. Central Database and Diet Tracker

Data transmitted from the ESP32 is ingested by a FastAPI backend connected to a PostgreSQL database. The application features a comprehensive diet tracker where users log their daily food intake. The system calculates macronutrients (proteins, carbohydrates, fats) and total caloric intake, storing this alongside the user's fixed biometrics (age, weight, and pre-calculated stride length).

IV. METHODOLOGY

1. Step and Speed Calculation

When a user taps their RFID card, the ESP32 begins polling the MPU-9250 at 50 Hz. We apply a moving average filter to the Z-axis accelerometer data to isolate the distinct physical shock of a footfall from ambient machine vibrations. A step is counted when the Z-axis acceleration exceeds a dynamically calibrated threshold, Th_z.

During the onboarding phase in the FitSense app, the user's stride length (Ls) is calculated based on their height. The distance (D) covered is calculated as:

$$D = Nsteps \times Ls$$

where Nsteps is the total steps counted. The average speed is subsequently derived by dividing D by the session duration measured between the RFID tap-in and tap-out.

Table 1. Sensor Configuration

Component	Function	Interface
ESP32	Main controller, Wi-Fi	-
RC522	User authentication	SPI
MPU-9250	Vibration/Step tracking	I2C

To prevent overtraining, an algorithmic guardrail is enforced post-prediction: the recommended workout load cannot exceed 115% of the user's 10-session rolling average, irrespective of the ML output.

2. The Unified Recommendation Engine

The core innovation of FitSense is its holistic ML engine. We deploy a Multi-Output Random Forest Regressor [5], chosen for its ability to handle non-linear tabular data efficiently. The model takes a comprehensive array of inputs bridging both diet and exercise.

Table 2. Ml Model Input Features

Feature	Description
age, weight	User biometrics
calorie_delta	Consumed vs. Target calories
protein_score	Adequacy of protein intake (0-100)
avg_speed_10	Average speed over last 10 sessions
session_duration	Average duration of last 10 sessions

The model outputs three critical recommendations for the next day: Target Duration, Target Speed, and a Suggested Caloric Goal. If a user is consistently under-eating (negative calorie_delta), the engine will dynamically reduce the recommended workout intensity and prompt the diet tracker to suggest high-protein, calorie-dense meal options to aid recovery.

V. RESULTS AND DISCUSSION

1. Hardware and Sensing Accuracy

Initial testing of the ESP32 architecture demonstrated significant improvements in deployment scalability. By removing the dependency on individual Raspberry Pi gateways for each machine, the cost per node was reduced by approximately 60%. The MPU-9250 step inference algorithm, when tested against manual counting, achieved a 92% accuracy rate for walking speeds (4-6 km/h) and an 88% accuracy rate for running speeds (>8 km/h), as higher speeds introduced more complex chassis resonance.

2. System Latency

The RFID authentication pipeline is highly responsive. The time from a user tapping the RC522 to the session initiating on the backend averages 450 ms over a standard 2.4 GHz Wi-Fi network. Data payloads are sent every 5 seconds, containing aggregated step counts to minimize network congestion.

3. Ecosystem Efficacy

In simulated testing using synthetic profiles, the closed ecosystem successfully modulated exercise plans based on diet. Profiles simulating severe caloric deficits correctly received lower intensity recommendations, demonstrating the value of integrating nutritional data directly into the equipment tracking loop.

VI. CONCLUSION

FitSense successfully demonstrates a closed, holistic IoT ecosystem for gym environments. By shifting the sensing methodology to frame vibrations using an ESP32 and MPU-9250, and utilizing an RC522 RFID reader for session bounding, we create a highly scalable tracking solution. More importantly, integrating this hardware with a unified diet and exercise platform allows our Random Forest ML engine to generate safe, personalized recommendations that respect the user's actual physical exertion and nutritional intake. Future work will focus on extensive human trials to refine the step-inference algorithm across different machine types, such as ellipticals and stair climbers, and expanding the dietary database.

REFERENCES

1. T. J. Gabbett, "The training-injury prevention paradox: should athletes be training smarter and harder?," Br. J. Sports Med. , vol. 50, no. 5, pp. 273-280, 2016.
2. M. D. Mifflin et al., "A new predictive equation for resting energy expenditure in healthy individuals," Am. J. Clin. Nutr. , vol. 51, no. 2, pp. 241-247, 1990.
3. InvenSense Inc., "MPU-9250 Product Specification," Datasheet.

4. NXP Semiconductors, "MFRC522: Standard performance MIFARE and NTAG frontend," Datasheet.
5. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res. , vol. 12, pp. 2825-2830, 2011.
6. Sharjil Iqbal. "NETWORK ATTACK DETECTION BY GROUP SEARCH OPTIMIZATION USING CONVOLUTIONAL DEEP LEARNING MODEL