International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Application for Detection of Sleep Apnea using Machine Learning

Mrs. Ashwini Ramegowda[1], Sanjan M L[2], Sathvik H N[3], Shivu K[4], Nikhil Patil[5]

[1]*Ashwini Ramegowda, Asst. Prof, Dept. of ISE, East West Institute of Technology*

[2]*Sanjan M L, Dept. of ISE, East West Institute of Technology*

[3]*Sathvik H N, Dept. of ISE, East West Institute of Technology*

[4]*Shivu K, Dept. of ISE, East West Institute of Technology*

[5]*Nikhil Patil, Dept. of ISE, East West Institute of Technology*

---***---

Abstract - Sleep apnea is a disorder characterized by repeated drops in oxygen saturation and fluctuations in heart rate during sleep. Clinical diagnosis through polysomnography is accurate but expensive, time-consuming, and not easily accessible for routine screening. This project presents a portable and affordable sleep apnea monitoring system that uses a pulse oximeter sensor interfaced with a microcontroller to continuously measure SpO_2 and heart rate. The microcontroller collects real-time physiological data, processes the signals, and identifies potential apnea events based on oxygen desaturation patterns and abnormal pulse-rate variations. The system stores the readings for later evaluation and provides a simple method for preliminary detection without the need for complex clinical equipment. Experimental results indicate that monitoring these two key parameters using a pulse oximeter is sufficient to capture physiological changes commonly associated with sleep apnea. The proposed design offers a practical and low-cost alternative for home-based screening and early identification of sleep-related breathing disorders.

Key Words: Sleep Apnea, SpO_2, Heart rate, Micro controller, Pulse Oximeter

1.INTRODUCTION

Sleep apnea is a common yet serious sleep disorder in which breathing repeatedly stops and starts during sleep, leading to fatigue and long-term health risks such as heart disease and hypertension. This project proposes a sensor-based and machine-learning approach to detect sleep apnea using two key physiological parameters: heart rate and SpO_2. These signals are collected through a pulse oximeter sensor, enabling continuous and non-invasive monitoring during sleep. By employing the Random Forest algorithm, the system classifies and predicts the likelihood of sleep apnea with high accuracy based on variations in oxygen saturation and pulse rate. In addition to sleep apnea, related sleep disorders such as insomnia, parasomnia, and snoring play an important role in sleep quality assessment. Insomnia frequently occurs in patients with Obstructive Sleep Apnea (OSA), with nearly half experiencing moderate to severe symptoms. Parasomnia involves abnormal physical behaviors during sleep that may resemble psychiatric conditions. Snoring, caused by obstructed airflow, often serves as an early indicator of sleep apnea. Despite the significant impact of these disorders, many individuals remain undiagnosed because traditional methods—such as polysomnography—are expensive, uncomfortable, and require specialized clinical environments.

2. Body of Paper

2.1 System Overview

The proposed system aims to detect sleep apnea by continuously monitoring two physiological parameters: blood oxygen saturation (SpO_2) and heart rate. These parameters are captured using a pulse oximeter sensor connected to a microcontroller, which handles data acquisition, preprocessing, and transmission. A machine learning model (Random Forest) analyzes the recorded data to identify patterns associated with apnea events. The system is designed to provide a low-cost, non-invasive, and home-friendly screening method for sleep apnea detection.

2.2 Hardware Components

The hardware consists of two main elements:
Pulse Oximeter Sensor:
This sensor measures SpO_2 and heart rate using photoplethysmography. It detects variations in blood volume changes through light absorption, enabling real-time physiological monitoring.
Microcontroller: The microcontroller collects the sensor readings, processes the signals, performs basic filtering, and sends the processed data to storage or a classifier. It also ensures stable sampling during sleep without disturbing the user. This minimal hardware design ensures portability, affordability, and continuous overnight monitoring.

2.3 Data Collection and Preprocessing

The pulse oximeter generates raw physiological data containing possible noise due to body movement or sensor shifts. The microcontroller applies preprocessing methods such as:

- Noise filtering: Removes spikes and sudden artefacts in the PPG signal.
- Smoothing: Uses moving averages to refine heart rate and SpO_2 values.
- Normalization: Ensures consistent input ranges for machine learning.

After preprocessing, structured data is stored as time-series samples for further analysis.

2.4 Machine Learning Model

A Random Forest classifier is employed to detect probable apnea events. The model uses features derived from SpO_2 and heart-rate changes to classify breathing states. Random Forest is chosen because:

- It is robust to noise and variable data
- It performs well with small feature sets
- It handles nonlinear relationships effectively

The model is trained with labeled datasets and later used to predict apnea-like events during testing.

2.5 System Design

The system is designed with a simple yet efficient architecture combining hardware sensing and software analysis. The hardware layer collects physiological signals, while the software layer processes and analyses.

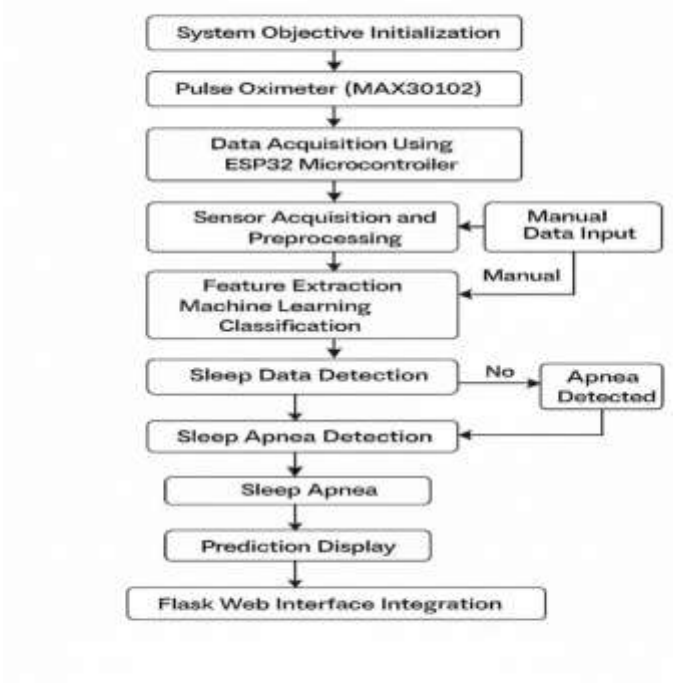


Fig -2.1: System Architecture

- The pulse oximeter measures SpO_2 and heart rate.
- The microcontroller handles real-time data sampling and preprocessing.
- Preprocessed data is fed to the machine-learning classifier.
- Detected apnea events are logged and analyzed.

This layered design ensures accurate data capture, reliable processing, and effective apnea detection with minimal hardware complexity.

2.6 Methodology

The methodology includes the following steps:

1. The process starts with physiological signal acquisition, where the pulse oximeter continuously measures SpO_2 and heart rate using PPG and sends these readings to the ESP32.
2. The ESP32 performs data preprocessing, including noise filtering, smoothing, and basic validation, so that motion artefacts and unstable readings are reduced before analysis.
3. The preprocessed SpO_2 and heart-rate values are then provided to the ML classifier, which uses a trained Random Forest model to analyse desaturation patterns and heart-rate fluctuations and classify whether the pattern is normal or apnea-like.
4. The classification results, along with the corresponding sensor data, are stored in a database for long-term logging, trend analysis, and future model improvement.
5. When the ML classifier detects abnormal or potentially dangerous conditions, the ESP32 triggers an alert module, which generates notifications to the user or caregiver about the abnormal event.
6. At the same time, live readings and prediction outputs are sent to a web dashboard, where users can view real-time SpO_2 and heart-rate graphs, as well as apnea predictions and historical records.

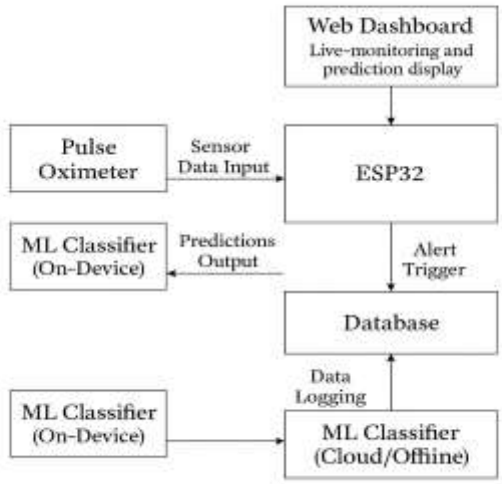


Fig -2.2: Methodology

2.7 Implementation

The implementation is divided into hardware interfacing and machine-learning execution.

2.7.1 Hardware Implementation:

- **ESP32 Microcontroller**:
  ESP32 is a low-power Wi-Fi-enabled microcontroller with dual-core architecture,

suitable for IoT applications. It communicates with sensors and sends processed data to the server.

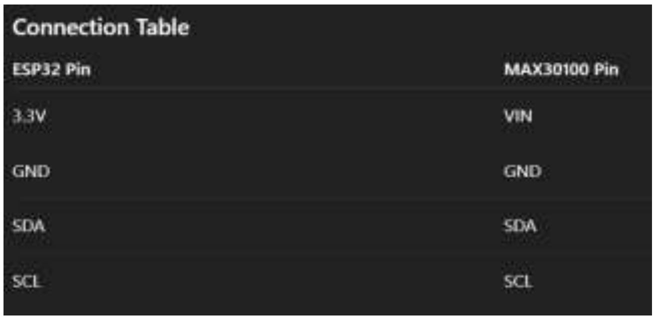
Fig -2.3: ESP32 Microcontroller

- **MAX30100 Pulse Oximeter Sensor:**

IR LED & Red LED
Photodiode
Heart rate engine
SpO_2 processing unit

Fig -2.4: MAX30100 Oximeter

- **Hardware Connections:**



Connection Table

ESP32 Pin	MAX30100 Pin
3.3V	VIN
GND	GND
SDA	SDA
SCL	SCL

Fig -2.5: Connection Table

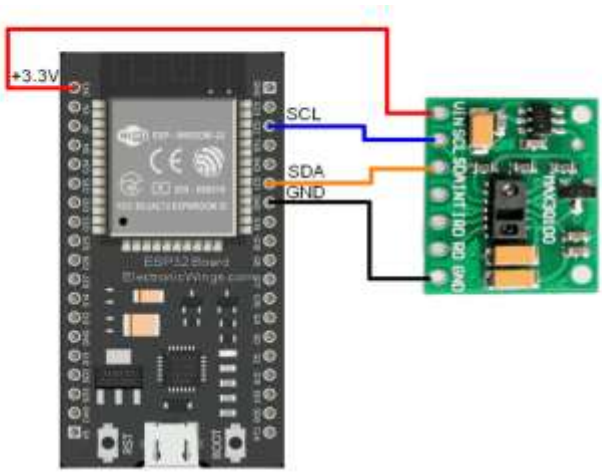
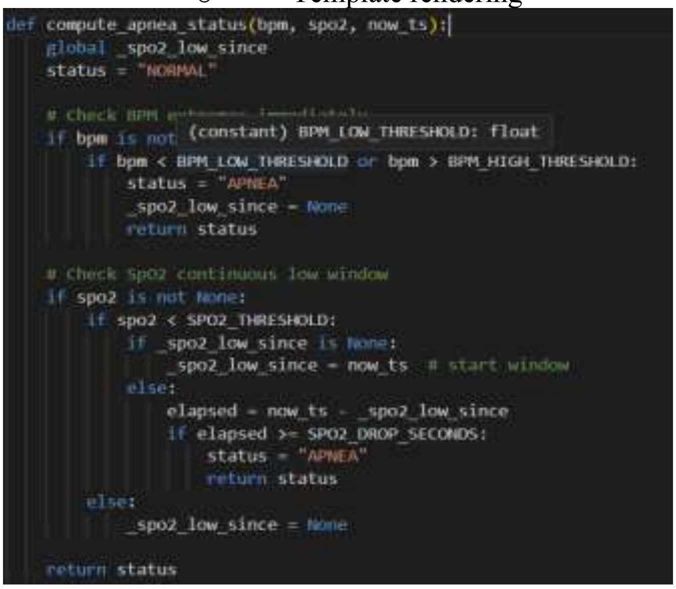


Fig -2.6: Connection Photo

2.7.2 Software Implementation:

- **Backend Implementation(Flask):**
  - Serial thread
  - Login/Registration
  - Data streaming (SSE)
  - Prediction route
  - Template rendering



```
def compute_apnea_status(bpm, spo2, now_ts):
    global _spo2_low_since
    status = "NORMAL"

    # Check BPM extremes immediately
    if bpm is not (constant) BPM_LOW_THRESHOLD: float
        if bpm < BPM_LOW_THRESHOLD or bpm > BPM_HIGH_THRESHOLD:
            status = "APNEA"
            _spo2_low_since = None
            return status

    # Check SpO2 continuous low window
    if spo2 is not None:
        if spo2 < SPO2_THRESHOLD:
            if _spo2_low_since is None:
                _spo2_low_since = now_ts  # start window
            else:
                elapsed = now_ts - _spo2_low_since
                if elapsed >= SPO2_DROP_SECONDS:
                    status = "APNEA"
                    return status
        else:
            _spo2_low_since = None

    return status
```

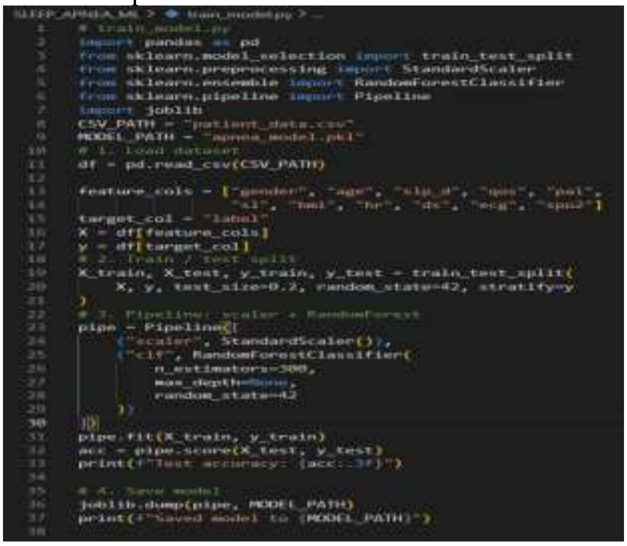
Fig -2.7: Sleep Apnea Status

- **Machine Learning Model**

Model : Random Forest Classifier
Features Used (11):

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 12 | Dec - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

gender, age, slp_d, qos, pal, sl, bmi, hr, ds, ecg, spo2.



```python
# train_model.py
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.pipeline import Pipeline
import joblib
CSV_PATH = "patient_data.csv"
MODEL_PATH = "apnea_model.pkl"
# 1. Load dataset
df = pd.read_csv(CSV_PATH)

feature_cols = ["gender", "age", "slp_d", "qos", "pal",
                "sl", "bmi", "hr", "ds", "ecg", "spo2"]
target_col = "label"
X = df[feature_cols]
y = df[target_col]
# 2. Train / test split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
# 3. Pipeline: scaler + RandomForest
pipe = Pipeline([
    ("scaler", StandardScaler()),
    ("clf", RandomForestClassifier(
        n_estimators=300,
        max_depth=None,
        random_state=42
    ))
])
pipe.fit(X_train, y_train)
acc = pipe.score(X_test, y_test)
print(f"Test accuracy: {acc:.3f}")

# 4. Save model
joblib.dump(pipe, MODEL_PATH)
print(f"Saved model to {MODEL_PATH}")
```

Fig -2.8: Train Model

2.8 Results

The system was tested by continuously monitoring SpO_2 and heart rate over multiple sleep intervals, and the collected data was processed in real time through the ESP32 and the machine-learning classifier. The recorded SpO_2 trends showed noticeable drops during periods where apnea-like events were expected, while heart-rate variations displayed corresponding fluctuations, confirming the physiological relationship between desaturation and cardiovascular response. The Random Forest classifier demonstrated reliable performance in identifying these abnormal patterns, producing accurate predictions while minimizing false detections. The system successfully logged all readings to the database and displayed real-time graphs on the web dashboard, allowing users to visualize both live data and predicted events. Additionally, the alert module generated notifications whenever abnormal SpO_2 levels were detected, confirming the responsiveness of the system in real-time monitoring conditions. Overall, the results indicate that the proposed system is effective for preliminary sleep-apnea screening and provides dependable continuous monitoring using minimal hardware.

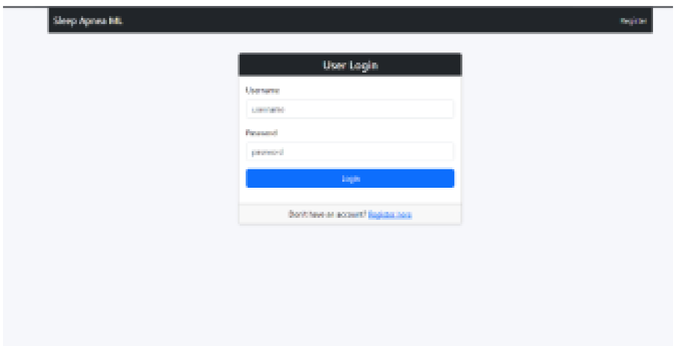


Fig -2.9: Login Page

Fig -2.10: Manual Prediction

Fig -2.11: Live Monitor

3. CONCLUSIONS

The proposed system successfully demonstrates a practical and cost-effective solution for preliminary detection of sleep apnea using only SpO_2 and heart-rate measurements. By integrating a pulse oximeter with an ESP32 microcontroller and a Random Forest classifier, the system is able to analyze physiological variations in real time and identify apnea-like events with reliable accuracy. The addition of a web dashboard provides continuous live monitoring and easy visualization of trends, while the alert module ensures immediate notification during abnormal conditions. Experimental results confirm that the system operates efficiently and responds effectively to variations in oxygen saturation and heart rate. Although not a replacement for clinical polysomnography, this approach offers a promising, accessible, and user-friendly method for home-based screening and early detection of sleep-related breathing disorders.

REFERENCES

- **Alireza Pedram Razi, and Zahra Einalou**, "Sleep Apnea Classification Using Random Forest via ECG," International Journal of Medical Engineering, 2023.
- **Anju Prabha**, "Respiratory Effort Signal Based Sleep Apnea Detection System Using Improved Random Forest Classifier," Biomedical Signal Processing Letters, 2022.
- **Kandala N.V.P.S. Rajesh**, "Obstructive Sleep Apnea Detection Using Discrete Wavelet

Transform-Based Statistical Features," Journal of Healthcare Technologies, 2022.

- **Rahman, A., & Gupta, M.**, "Non-Invasive Sleep Apnea Detection Using SpO_2 and Heart Rate Signals with Machine Learning," International Journal of Biomedical Engineering and Technology, 2023.
- **S. Kumar, and Harini, P.**, "IoT-Based Health Monitoring System Using ESP32 and MAX30102 Pulse Oximeter Sensor," Journal of Sensor Networks and Applications, 2023.
- **M. Rakesh, and A. Deepak**, "Real-Time SpO_2 and Heart Rate Measurement Using MAX30102 and ESP32 Microcontroller," Embedded Systems Review, 2022.
- **P. Chatterjee, and R. Sen**, "A Flask-Based IoT Framework for Real-Time Physiological Data Monitoring," International Conference on IoT, Data Science and Computing Applications, 2024.
- **MAX30102 Datasheet**, "MAX30102 Pulse Oximeter and Heart-Rate Sensor Module Technical Documentation," Maxim Integrated, 2021.
- **Espressif Systems**, "ESP32 Technical Reference Manual," Espressif, 2022.