Health Trends Analysis Using Wearable Device Data

Hithaishi Raghavendra Reddy
raghavendrareddy.h@northeastern.edu
Northeastern University
Boston, Massachusetts

Tanvi Prashant Magdum
magdum.t@northeastern.edu
Northeastern University
Boston, Massachusetts

ACM Reference Format:
Hithaishi Raghavendra Reddy and Tanvi Prashant Magdum. 2024. Health Trends Analysis Using Wearable Device Data. In . ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In today's fast-paced academic environment, particularly for graduate students in Computer Science and Data Science, the demands of coursework, exams, and project deadlines can lead to heightened stress and compromised physical health. Many students find themselves prioritizing their academic obligations over their well-being, often neglecting regular exercise, quality sleep, and monitoring their physical health. This oversight, though common, can result in fatigue, decreased energy levels, and a general decline in overall health, further impacting academic performance. Unfortunately, despite the growing availability of wearable devices that track health metrics, most students are unaware of how to make sense of the data these devices collect, or how to use it to improve their overall health.

This project aims to bridge that gap and focuses on analyzing data from wearable devices, specifically daily activity, sleep patterns, calories burned, and heart rate fluctuations, to uncover trends and make predictions that could enhance students' health. By examining how factors like physical activity, sleep, and calorie intake correlate, we can uncover insights that may help students manage their stress better and develop healthier habits. For example, we might discover that students who engage in regular physical activity tend to sleep better and have lower resting heart rates, which can lead to increased energy levels and improved focus for studying. These findings could also help predict which students might be at risk of burnout or poor health due to irregular activity or sleep patterns.

The main goal of this analysis is to highlight the importance of paying attention to even the small aspects of our health and how they can impact our academic performance and overall well-being. By developing predictive models, we aim to provide real-life recommendations for students to improve their physical and mental health. These insights could guide better decision-making, such as setting fitness goals based on personal activity patterns, adjusting sleep routines for optimal recovery, and even predicting when additional rest is needed to avoid burnout. Ultimately, this project demonstrates how data-driven insights can help students balance academic demands with self-care, creating a more sustainable and healthier lifestyle.

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 REVIEW OF LITERATURE

Chandrasekaran et al. (2020) [5] analyzed the adoption and predictors of wearable health device usage among US adults using a national survey of 4551 respondents. The study revealed that only 30 percent of US adults use wearable devices, with a greater prevalence among younger, educated, and affluent individuals. Factors such as higher technology self-efficacy, positive health perceptions, and an active lifestyle were significantly associated with increased usage. The findings highlighted a digital divide, with older and less affluent populations underrepresented in wearable device usage, emphasizing the need for targeted interventions to bridge these gaps.

Loncar-Turukalo et al. (2019) [6] conducted a scoping review of wearable health technology from 2010 to 2019, identifying key trends, milestones, and barriers. While advancements in miniaturization, 5G communications, and edge computing have driven technological progress, challenges such as energy efficiency, privacy concerns, and user acceptance persist. The study suggested that addressing these barriers requires a multidisciplinary approach involving technology innovation, stakeholder collaboration, and formalized use-case development to ensure widespread adoption.

Lin Lu et al. (2020) [3] reviewed 82 studies on the use of wearable devices in clinical settings, classifying their applications into four areas: health monitoring, chronic disease management, diagnosis and treatment, and rehabilitation. Despite their promise, the authors identified limitations, including user-friendliness, privacy concerns, and technical bottlenecks, which hinder broader integration into clinical workflows. The review underscored the need for industry standards and research into advanced applications to optimize wearable device use in health care.

Conger et al. (2021) [1] examined trends in physical activity levels using data from wearable devices over a 22-year period. The meta-analysis revealed significant declines in activity levels, particularly among adolescents. These findings highlighted the importance of wearables in tracking long-term physical activity trends and underscored the need for targeted strategies to promote physical activity, especially in youth populations.

Zhu et al. (2020) [2] proposed a framework leveraging wearable device data to predict COVID-19 epidemic trends. Using heart rate and sleep data from 1.3 million users, the study developed a neural network model that combined physiological anomaly rates with historical infection data. The framework demonstrated potential in providing early warnings of outbreaks, showcasing the role of wearable devices in public health surveillance and disease prevention.

Soh et al. (2021) [4] explored the technical challenges and future trends in wearable health-monitoring systems. The study highlighted issues such as interference with user movements, electromagnetic safety, and material durability as significant design considerations. Advances in body-conformal sensors and reliable wireless communication are critical for the next generation of wearable systems to overcome these challenges and enhance their practical utility.

3 METHODOLOGY

This project employs a quantitative research methodology, which involves the collection and analysis of numerical data to identify patterns and trends. The purpose of this approach is to apply statistical and computational techniques to draw meaningful insights from the wearable device data, specifically focusing on activity, sleep, and heart rate. By analyzing these metrics, the goal is to make data-driven conclusions about individual health behaviors and develop predictive models based on historical trends. The methodology includes data collection, cleaning, pre-processing, and analysis, with the primary goal of uncovering patterns and building visualizations to support actionable health insights.

3.1 Data Collection

The dataset used in this project was sourced from Kaggle, containing activity, sleep, and heart rate data collected through wearable fitness devices (Fitbit). While the original dataset included a wide array of parameters, only the necessary fields sufficient for this study were selected. The data was recorded for 33 individuals over a one-month period, from April 12, 2016, to May 12, 2016, capturing various fitness and health metrics.

The daily activity dataset contains records of physical activity metrics such as total steps, distance traveled, active minutes, and calories burned. The sleep dataset provides information on sleep patterns, including total sleep duration and time spent in bed. Lastly, the heart rate dataset records heart rate values at second-level granularity.

These datasets, provided in CSV format, were imported into Python for further processing and analysis. They offer a comprehensive view of participants' physical activities, sleep patterns, and heart rate fluctuations over the study period, forming the basis for all subsequent analyses.

3.2 Data Cleaning

To ensure data reliability, we performed data cleaning to remove the inconsistencies. For all datasets, extra spaces in column names were stripped to standardize naming conventions. Date fields, such as 'ActivityDate' in the daily activity dataset and 'SleepDay' in the sleep dataset, were converted to *datetime* formats, allowing for time-based analyses. Duplicate and missing records were removed across all datasets to avoid redundancy and preserve data integrity.

For the heart rate dataset, once the dataset was resampled, cleaning involved filtering data to include only active hours (from 7:00 AM to 8:00 PM) and interpolating missing values to maintain continuity in trends. This step was crucial for meaningful time-series analysis, as missing or irrelevant entries could distort patterns.

3.3 Data Preprocessing

Once cleaned, the data underwent preprocessing to make it suitable for analysis. In the daily activity dataset, numerical columns such as 'TotalSteps' and 'Calories' were examined for outliers using the interquartile range (IQR) method. Records with extreme values were removed to minimize their influence on statistical summaries and visualizations. The activity minutes were aggregated into a new feature - 'TotalActiveMinutes', combining very active, fairly active, and lightly active minutes to represent total daily activity comprehensively.

We applied normalization to numerical columns in the daily activity dataset using Min-Max Scaling. This ensured that variables were scaled within a consistent range, particularly useful for comparing metrics like steps, distance, and calories across different magnitudes.

For the sleep dataset, sleep duration in minutes was converted into hours for easier interpretation, and an 'AverageSleepDuration' feature was calculated.

In the heart rate dataset, we calculated hourly averages from second-level data through resampling, significantly reducing the dataset size while retaining critical information. Lag features were engineered to incorporate prior heart rate values, enabling predictive analysis.

3.4 Data Analysis

We explored activity trends, sleep patterns, and heart rate fluctuations using advanced techniques. We computed aggregated metrics to examine weekday trends in activity levels, calories burned, and sleep duration for all the individuals. We generated correlation matrices for the daily activity dataset, revealing relationships among metrics such as total steps and calories burned.

We utilized K-Means clustering for the sleep dataset to identify distinct sleep behavior groups, such as individuals with short, average, or extended sleep durations. These clusters provided insights into variability in sleep habits across the population.

In the heart rate dataset, we performed predictive modeling using linear regression and random forest regression to forecast heart rate fluctuations and compare the accuracy. Lagged heart rate values were used as predictors, enabling the identification of patterns and trends in heart rate variability. Model performance was evaluated using metrics such as root mean squared error (RMSE) and mean absolute percentage error (MAPE), with the random forest model achieving slightly higher accuracy than linear regression model.

This structured approach to data analysis enabled a comprehensive understanding of physical activity, sleep behaviors, and physiological responses, offering actionable insights for optimizing health and fitness.

3.5 Data Visualization

We have developed a data-driven web application to analyze heart rate fluctuations, trends, and sleep patterns using Flask for the backend and React for the frontend. The backend integrates Flask with the 'plotly' library for interactive visualizations, while the frontend uses Material-UI components and React Router for navigation. The backend logic processes a dataset of heart rate readings, resampled to hourly averages, and applies machine learning models such as Linear Regression and Random Forest to predict future heart rates.

For trends and sleep analysis, additional datasets including daily activity and sleep data are preprocessed through steps like duplicate removal, outlier detection, and feature scaling using 'MinMaxScaler'. Data visualization is accomplished through libraries like Seaborn and Matplotlib for cluster analysis and insights.

The frontend incorporates tabs for navigating between "Heart Rate Analysis" and "Trends" sections. A custom 'Navigation' component built with Material-UI enables smooth transitions, while 'CustomTabs' enhances user experience with visually styled tabs for modular analysis. The 'HeartRateAnalysis' component implements a user-friendly interface for submitting user IDs and fetching plots. Interactive design elements like animated transitions, form validation, and loading indicators are included using Framer Motion and Material-UI.

4 RESULTS

4.1 Activity Trends Analysis

4.1.1 Background. Understanding daily activity trends is crucial for fostering a healthy lifestyle. By analyzing data such as step counts, active minutes, and calories burned across different days and times, we can identify patterns that help optimize fitness routines. Activity patterns often vary due to work schedules, leisure activities, or exercise habits, and observing these trends provides valuable insights into how activity is distributed over the week. Investigating correlations between activity metrics like total active minutes and calories burned helps understand how effectively individuals utilize their physical activity for energy expenditure. This analysis also highlights individual behaviors, enabling personalized recommendations for achieving health goals.

4.1.2 Interpretation.

(1) **Average Activity Metrics Across Weekdays**: This plot reveals a largely consistent trend in physical activity during the week, with total steps, calories burned, and active minutes maintaining a steady range. 'TotalSteps' typically range between 6,000 and 8,000, 'Calories' vary from 2,100 to 2,300, and 'TotalActiveMinutes' fall between 199 and 240. Tuesdays show a noticeable peak in step count, suggesting heightened activity on this day, possibly due to mid-week fitness routines or outdoor activities. In contrast, Sundays display a significant dip in step count, indicating reduced physical activity, likely due to rest or sedentary leisure. These patterns suggest a stable level of activity during the workweek but highlight opportunities to incorporate light exercises on Sundays to maintain consistency and improve overall health.

(2) **Correlation Heatmap of Activity Metrics**: This heatmap provides a comprehensive view of the relationships between various physical activity parameters. Strong positive correlations are evident between 'TotalSteps' and 'TotalDistance', as

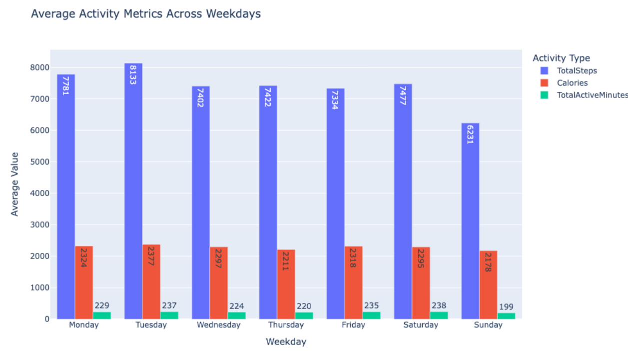

Figure 1: Average Activity Metrics Across Weekdays

expected, since more steps directly translate into greater distance covered. Similarly, 'Calories' show a robust correlation with 'VeryActiveMinutes' and 'TotalSteps', indicating that higher-intensity activities contribute significantly to calorie expenditure. 'LightlyActiveMinutes' exhibit moderate correlations with parameters like 'Calories' and 'TotalSteps', reflecting their impact on overall activity levels despite being less intense. In contrast, 'SedentaryMinutes' show a negative correlation with most activity metrics, underscoring the inverse relationship between inactivity and physical exertion. These insights emphasize the importance of maximizing active minutes, particularly vigorous activities, to improve calorie burn and overall fitness, while minimizing sedentary periods to enhance daily activity balance.

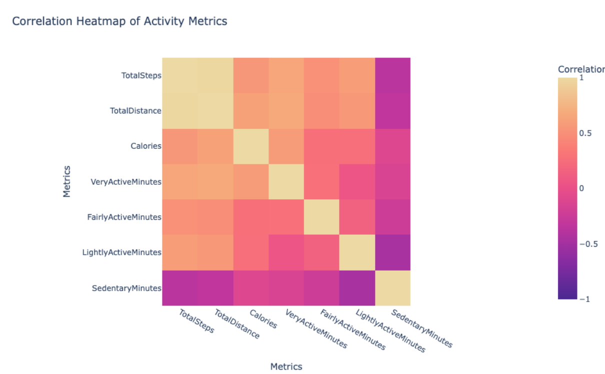

Figure 2: Correlation Heatmap of Activity Metrics

(3) **Calories Burned vs. Total Active Minutes**: This scatter plot confirms that higher active minutes lead to more calories burned, although individual variations exist due to factors like weight and metabolism. Points in the lower-left corner indicates sedentary behavior, signaling opportunities for improvement. Clustering is observed more in the upper-right corner indicating a good health routine for aggregated users.

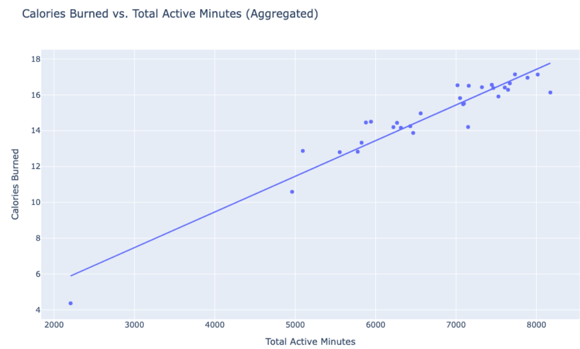

Figure 3: Calories Burned vs. Total Active Minutes

(4) **Activity Trends Over Time One User**: This time-series analysis captures day-to-day variability for one user with ID - 1503960366. It reveals patterns in the relationship between 'TotalSteps' and 'Calories' burned over a one-month period. Calories closely mirror the trends of total steps, with a consistent steps to calories ratio of approximately 3:2, emphasizing the direct link between physical movement and energy expenditure. During the observed time frame, activity levels peak between April 24 and May 1, suggesting a period of heightened physical engagement. However, there is a noticeable decline in activity as May progresses, potentially indicating reduced consistency or external factors affecting routine. The sharp fluctuations in the graph highlight varying daily activity levels, with Sundays consistently reflecting a drop in both steps and calories, possibly due to reduced activity on rest days. These patterns underscore the importance of maintaining steady physical activity throughout the week for sustained health benefits.

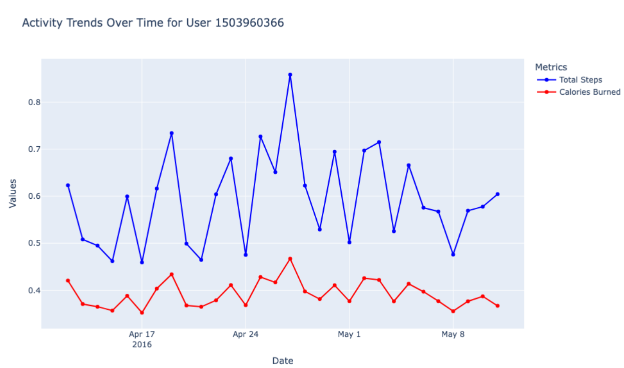

Figure 4: Activity Trends Over Time (Specific User)

4.2 Sleep Patterns Analysis

4.2.1 Background. Sleep quality plays a fundamental role in overall health, affecting cognition, mood, and physical recovery. By analyzing sleep patterns over time, we can derive actionable insights into improving sleep hygiene. Patterns such as sleep consistency (going to bed and waking up at similar times) and interruptions during sleep are critical in understanding factors that impact sleep quality. Visualizing these patterns allows for identifying habits or external factors that disrupt healthy sleep cycles.

4.2.2 Interpretation.

(1) **Sleep Patterns Over a Month (Aggregate)**: The scatter plot highlighting the relationship between Total Time in Bed and Total Hours Asleep provides a detailed view of sleep behavior across three clusters: Good Sleepers, Most Common Sleepers, and Inconsistent Sleepers. Good Sleepers exhibit a wider distribution, indicating variability in sleep quality. Despite spending sufficient time in bed, some individuals in this group do not achieve optimal sleep durations. This suggests potential disturbances such as difficulty falling asleep, frequent waking, or poor sleep hygiene. For these individuals, improving sleep efficiency through better routines, relaxation techniques, or addressing potential sleep disorders could prove beneficial.
Most Common Sleepers display a strong linear relationship between time in bed and sleep duration. Their data points closely align with the diagonal, suggesting a consistent balance between time spent in bed and actual sleep. This group represents individuals with stable and healthy sleep patterns who should continue maintaining their routines.
Inconsistent Sleepers show a similar linear trend but with overall lower sleep durations. This indicates irregular sleep patterns, insufficient sleep hours, or disruptions in their routines. Addressing these issues with strategies like increasing time in bed, reducing schedule inconsistencies, or prioritizing rest is essential for this group.

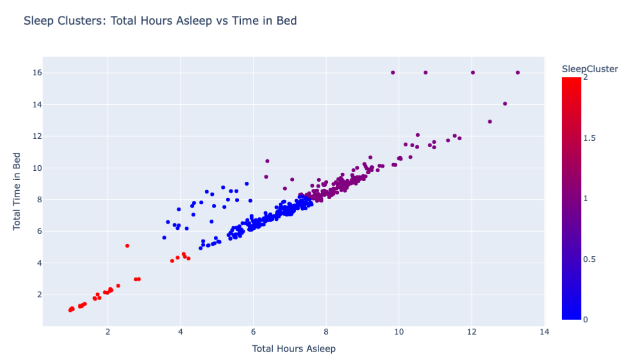

Figure 5: Sleep Patterns Over a Month (Aggregate)

(2) **Average Sleep Metrics by Cluster**: This bar plot compares average metrics such as Total Hours Asleep, Time in Bed, and Sleep Duration for each cluster further reinforces these insights. Good Sleepers shows higher average time in bed compared to sleep duration, highlighting inefficiencies. The gap between these metrics underscores the need to improve the quality of their sleep rather than just increasing the time

in bed. Most Common Sleepers demonstrates near-equal averages for time in bed and sleep duration, reflecting effective and balanced sleep habits. These individuals spend optimal time resting and appear to achieve the recommended 7–9 hours of sleep consistently. Inconsistent Sleepers displays the lowest averages across all metrics, with a little disparity between time in bed and actual sleep. This emphasizes the need for these individuals to prioritize sleep, as their patterns indicate potential deprivation or inconsistency.

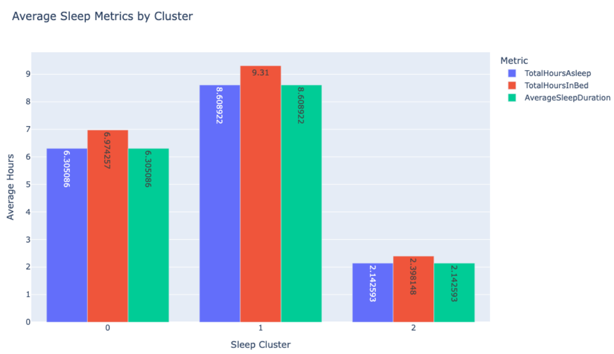

Figure 6: Average Sleep Metrics by Cluster

4.3 Individual Heart Rate Forecasting

4.3.1 Background. Heart rate data offers significant insights into physical fitness, stress levels, and overall cardiovascular health. Variations in heart rate can indicate activity intensity, recovery states, or even underlying health conditions. By forecasting heart rate patterns, individuals can optimize workouts, manage stress, and track improvements in fitness over time. Continuous monitoring and visualization of fluctuations also provide a snapshot of daily exertion and recovery, enabling users to balance activity and rest effectively.

4.3.2 Choice of Forecasting Models. For heart rate forecasting, Linear Regression and Random Forest Regressor were chosen for their interpretability and ability to capture complex patterns, respectively. Linear Regression offers a straightforward approach to understanding the relationship between lagged features (heart rate from the past three hours) and the target variable (next hour's heart rate). On the other hand, Random Forest is an ensemble learning method that builds multiple decision trees to capture non-linear and intricate dependencies within the data. This dual approach allows us to compare the predictive accuracy of a simple model versus a more robust, non-linear model.

4.3.3 Training and Testing Dataset. The dataset consists of hourly average heart rate readings derived from raw second-by-second heart rate data. The data was cleaning and pre-processed using resampling and interpolation methods. Lagged values (from the previous three hours) were included as predictors, reflecting the temporal dependencies in heart rate changes. The data was split into training (80%) and testing (20%) sets, ensuring that the model is evaluated on unseen data. This setup ensures that the models learn patterns from historical data and generalize well to future predictions.

4.3.4 Model Accuracy. Initially, Linear Regression was applied as a baseline model. While it provided a reasonable accuracy, it struggled with capturing the non-linear dynamics inherent in heart rate fluctuations, such as the impact of sudden exertion or recovery phases. To address this limitation, a Random Forest Regressor was introduced, which handles non-linear relationships effectively. It reduces overfitting through averaging predictions from multiple decision trees and automatically accounts for interactions between features (e.g., how Lag_1 and Lag_2 collectively influence the target). As a result, Random Forest achieved slightly higher accuracy, showcasing its superior ability to predict heart rate trends.

4.3.5 Interpretation. The Random Forest model outperformed Linear Regression with a slightly higher accuracy for almost any user. This demonstrates the importance of selecting models that can account for the complexities of physiological data. We analyzed heart rate metrics like Average Heart Rate providing a baseline understanding of the user's cardiovascular activity during active hours, Resting Heart Rate estimated from the lowest 10% of readings, indicating recovery and overall heart health and Peak Heart Rate representing moments of intense exertion, useful for assessing fitness levels and ensuring activity does not exceed safe limits.

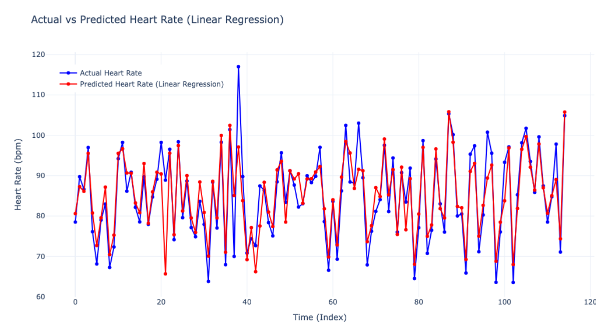

Figure 7: Linear Regression

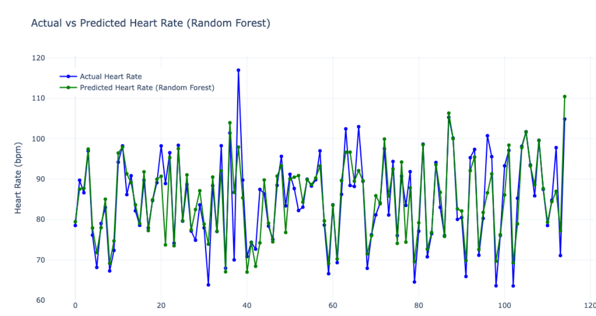

Figure 8: Random Forest

5 DISCUSSION

5.1 Limitation

5.1.1 Data Scope and Generalization. The dataset used for this analysis is limited to 33 participants and the data is recorded only for 1 month, which may not fully represent the variability in patterns across diverse populations, age groups, or health conditions. Heart rate data covers primarily active periods (7:00 am to 8:00 pm), which can overlook important information during resting or sleeping hours.

5.1.2 Modeling Limitations. Linear Regression is constrained by its assumption of linearity, which may oversimplify the relationship between lagged heart rate values and future fluctuations. Random Forest, while more flexible, lacks transparency in its decision-making process, making it harder to interpret individual predictions. Moreover, Random Forest models can overfit when trained on small datasets, as the set of trees may capture noise rather than meaningful patterns.

5.1.3 Temporal and Seasonal Factors. The heart rate analysis does not account for temporal variations (e.g., time of year or week) or individual-specific differences, such as fitness levels, sleep habits, or stress levels, which can influence heart rate dynamics.

5.1.4 Resolution of Data. Aggregating data to hourly averages is easier to work with and interpret, but it can smooth out short-term fluctuations that are critical for precise forecasting and real-time decision-making.

5.2 Future Research

Expanding the dataset to include more participants with diverse demographics, health statuses, and activity levels will improve generalizability. Incorporating nighttime data could provide a comprehensive view of heart rate trends. Adding external variables such as physical activity intensity, sleep quality, environmental factors (e.g., temperature), and emotional states could enhance the predictive accuracy and provide richer insights.

We can also explore advanced deep learning methods, such as Long Short-Term Memory (LSTM) networks as they have ability to handle time-series data with complex, non-linear dependencies. These models may provide more accurate predictions. Developing individualized models tailored to specific users' physiological patterns could yield more accurate and relevant forecasts. These models can be dynamically updated as new data becomes available. Further, future research could assess the impact of real-time feedback and heart rate forecasting on behavioral outcomes, such as exercise adherence or stress management.

6 CONCLUSIONS

This study utilized wearable device data to analyze key health aspects, including activity trends, sleep patterns, and heart rate fluctuations, offering meaningful insights into personal wellness. Through data visualization and predictive modeling, we demonstrated the potential of wearable technology in providing actionable recommendations for improving health and fitness. Each analysis section presented unique findings that can guide individuals in optimizing their daily routines and making informed lifestyle choices.

The analysis of activity trends revealed clear patterns in user behavior, such as variations in activity levels across weekdays and correlations between activity metrics like calories burned and active minutes. These findings emphasize the importance of maintaining consistent physical activity levels to enhance calorie expenditure and overall fitness. Furthermore, personalized activity tracking allows users to identify their most active periods, enabling targeted improvements in their exercise regimens.

In examining sleep patterns, the study highlighted the impact of sleep duration and efficiency on overall recovery and performance. By analyzing the relationships between sleep metrics, we identified trends that can help users adjust their sleeping habits for better health outcomes. The ability to track and interpret sleep data empowers individuals to prioritize quality sleep, a critical factor for physical and mental well-being.

The heart rate forecasting section showcased the value of predictive modeling in understanding cardiovascular health. Linear Regression and Random Forest models were employed to forecast heart rate fluctuations based on historical data. While both models demonstrated reasonable accuracy, Random Forest proved to be more effective due to its ability to capture non-linear patterns. These forecasts can be leveraged for real-time monitoring, allowing users to anticipate changes and adjust their activities accordingly.

This project highlights the growing potential of wearable devices combined with data science to enable proactive health management. By addressing limitations such as dataset size and model scalability in future research, these insights can be further refined to benefit a broader audience. Wearable technology, when coupled with advanced analytics, has the power to transform personal health monitoring, fostering better habits and improving long-term wellness.

REFERENCES

[1] L. P. TOTH C. CRETSINGER A. RAUSTORP J. MITÁŠ S. INOUE CONGER, S. A. and D. R. BASSETT. 2021. Time Trends in Physical Activity Using Wearable Devices: A Systematic Review and Meta-analysis of Studies from 1995 to 2017. (2021). https://exerciseismedicine.gr/wp-content/uploads/2022/03/Time_Trends_in_Physical_Activity_Using_Wearable.10.pdf

[2] Zi Meng Yi Yu Yanan Li Xiao Tang Yuling Dong Guangxin Sun Rui Zhou Hui Wang Kongqiao Wang Wang Huang Guokang Zhu, Jia Li. 2020. Learning from Large-Scale Wearable Device Data for Predicting the Epidemic Trend of COVID-19. (2020). https://doi.org/10.1155/2020/6152041

[3] Yi Xie Fei Gao Song Xu Xinghuo Wu Zhewei Ye Lin Lu, Jiayao Zhang1. 2020. (2020). https://mhealth.jmir.org/2020/11/e18907/

[4] Marco Mercuri Dominique M.M.-P. Schreurs Ping Jack Soh, Guy A.E. Vandenbosch. 2015. Wearable Wireless Health Monitoring: Current Developments, Challenges, and Future Trends. (2015). https://ieeexplore.ieee.org/abstract/document/7072588

[5] Evangelos Moustakas Ranganathan Chandrasekaran, Vipanchi Katthula. 2020. Patterns of Use and Key Predictors for the Use of Wearable Health Care Devices by US Adults: Insights from a National Survey. (2020). https://www.jmir.org/2020/10/e22443/

[6] José Machado da Silva Ioanna Chouvarda Vladimir Trajkovik Tatjana Loncar-Turukalo, Eftim Zdravevski. 2019. Literature on Wearable Technology for Connected Health: Scoping Review of Research Trends, Advances, and Barriers. (2019). https://www.jmir.org/2019/9/e14017

7 APPENDIX A CODE

```
import plotly.express as px
import plotly.graph_objects as go
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from flask import Blueprint, jsonify
import plotly.io as pio
```

```python

bp = Blueprint('trends', __name__)

@bp.route('/trends', methods=['GET'])
def display_trends():

    # Load data
    df_activity = pd.read_csv('dataset/new-data/dailyActivity.csv')

    # Clean data
    df_activity.columns = df_activity.columns.str.strip()
    df_activity['ActivityDate'] = pd.to_datetime(df_activity['ActivityDate'], errors='coerce')
    df_activity['Weekday'] = df_activity['ActivityDate'].dt.day_name()
    df_activity = df_activity.dropna().drop_duplicates()

    numerical_cols = ['TotalSteps', 'TotalDistance', 'Calories', 'VeryActiveMinutes', 'FairlyActiveMinutes', 'LightlyActiveMinutes', 'SedentaryMinutes']

    # Remove outliers for total steps
    Q1 = df_activity['TotalSteps'].quantile(0.25)
    Q3 = df_activity['TotalSteps'].quantile(0.75)
    IQR = Q3 - Q1
    df_activity = df_activity[~((df_activity['TotalSteps'] < (Q1 - 1.5 * IQR)) | (df_activity['TotalSteps'] > (Q3 + 1.5 * IQR)))]

    # Remove outliers for calories
    Q1 = df_activity['Calories'].quantile(0.25)
    Q3 = df_activity['Calories'].quantile(0.75)
    IQR = Q3 - Q1
    df_activity = df_activity[~((df_activity['Calories'] < (Q1 - 1.5 * IQR)) | (df_activity['Calories'] > (Q3 + 1.5 * IQR)))]

    # Combine activity levels to get TotalActivityMinutes
    df_activity['TotalActiveMinutes'] = df_activity['VeryActiveMinutes'] + df_activity['FairlyActiveMinutes'] + df_activity['LightlyActiveMinutes']

    # Visualization 1: Activity Distribution Over Weekdays
    weekday_activity = df_activity.groupby('Weekday')[['TotalSteps', 'Calories', 'TotalActiveMinutes']].mean().round()
    weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
    weekday_activity = weekday_activity.reindex(weekday_order)
    fig1 = px.bar(
        weekday_activity,
        title="Average Activity Metrics Across Weekdays",
        labels={'value': 'Average Value', 'variable': 'Activity Type'},
        barmode='group',
        text_auto=True
    )
    fig1.update_layout(xaxis_title='Weekday', yaxis_title='Average Value',
                       legend_title='Activity Type', width=1000, height=600 )
    fig1_json = pio.to_json(fig1)

    # Visualization 2: Heatmap of Activity Correlations
    corr_matrix = df_activity[numerical_cols].corr()
    fig2 = px.imshow(
        corr_matrix,
        title='Correlation Heatmap of Activity Metrics',
        labels=dict(x='Metrics', y='Metrics', color='Correlation'),
        x=numerical_cols,
        y=numerical_cols,
        color_continuous_scale='agsunset',
        zmin=-1,
        zmax=1
    )
    fig2.update_layout(width=1000, height=600)
    fig2_json = pio.to_json(fig2)

    # Initialize the scaler
    scaler = MinMaxScaler()

    # Normalization of Numeric Columns for Activity
    df_activity[numerical_cols] = scaler.fit_transform(df_activity[numerical_cols])

    # Visualization 3: Total Active Minutes vs. Calories (Aggregated)
    agg_data = df_activity.groupby('ActivityDate')[['TotalActiveMinutes', 'Calories']].sum().reset_index()
    fig3 = px.scatter(
        agg_data,
        x='TotalActiveMinutes',
        y='Calories',
        title='Calories Burned vs. Total Active Minutes (Aggregated)',
        labels={'TotalActiveMinutes': 'Total Active Minutes', 'Calories': 'Calories Burned'},
        trendline='ols'
    )
    fig3.update_layout(width=1000, height=600)
    fig3_json = pio.to_json(fig3)

    # Visualization 4: Activity Trends Over Time for a Single User
    user_id = df_activity['Id'].unique()[0]
    user_data = df_activity[df_activity['Id'] == user_id]
    fig4 = go.Figure()
    fig4.add_trace(
        go.Scatter(
            x=user_data['ActivityDate'],
            y=user_data['TotalSteps'],
            mode='lines+markers',
            name='Total Steps',
            line=dict(color='blue', width=2)
        )
    )
    fig4.add_trace(
        go.Scatter(
            x=user_data['ActivityDate'],
            y=user_data['Calories'],
            mode='lines+markers',
            name='Calories Burned',
            line=dict(color='red', width=2)
        )
    )
    fig4.update_layout(
        title=f'Activity Trends Over Time for User {user_id}',
        xaxis_title='Date',
        yaxis_title='Values',
        legend_title='Metrics',
        hovermode='x unified',
```

```
        width=1000,
        height=600
    )
    fig4_json = pio.to_json(fig4)

    return jsonify({"fig1": fig1_json, "fig2": fig2_json, "fig3": fig3_json, "fig4": fig4_json})
```

Listing 1: trends.py

```
import pandas as pd
import plotly.express as px
from sklearn.cluster import KMeans
from flask import Blueprint, jsonify
import plotly.io as pio

bp = Blueprint('sleepPatterns', __name__)

@bp.route('/sleeppatterns', methods=['GET'])
def display_sleep_patterns():

    # Load the dataset
    DailySleep = 'dataset/new-data/sleepDay.csv'

    df_sleep = pd.read_csv(DailySleep)

    # Data Cleaning
    df_sleep.columns = df_sleep.columns.str.strip()  # Remove extra spaces in column names
    df_sleep['SleepDay'] = pd.to_datetime(df_sleep['SleepDay'], errors='coerce')  # Convert to datetime
    df_sleep = df_sleep.dropna().drop_duplicates()  # Drop NA and duplicates

    # Calculate total sleep duration in hours
    df_sleep['TotalHoursAsleep'] = df_sleep['TotalMinutesAsleep'] / 60
    df_sleep['TotalHoursInBed'] = df_sleep['TotalTimeInBed'] / 60

    # Feature Engineering: Calculate Average Sleep Duration
    df_sleep['AverageSleepDuration'] = df_sleep['TotalMinutesAsleep'] / 60

    # Clustering Sleep Patterns (K-Means Clustering)
    kmeans = KMeans(n_clusters=3, random_state=42)
    df_sleep['SleepCluster'] = kmeans.fit_predict(df_sleep[['TotalHoursAsleep', 'TotalHoursInBed']])


    # Visualization 1: Sleep Clusters
    fig1 = px.scatter(
        df_sleep,
        x='TotalHoursAsleep',
        y='TotalHoursInBed',
        color='SleepCluster',
        title='Sleep Clusters: Total Hours Asleep vs Time in Bed',
        labels={'TotalHoursAsleep': 'Total Hours Asleep', 'TotalHoursInBed': 'Total Time in Bed'},
        color_continuous_scale='Bluered',
    )
    fig1.update_layout(xaxis_title='Total Hours Asleep', yaxis_title='Total Time in Bed',
                       legend_title='Sleep Cluster', width=1000, height=600 )
    fig1_json = pio.to_json(fig1)

    # Visualization 2: Average Sleep Metrics by Cluster
    sleep_summary = df_sleep.groupby('SleepCluster').agg({
        'TotalHoursAsleep': 'mean',
        'TotalHoursInBed': 'mean',
        'AverageSleepDuration': 'mean'
    }).reset_index()

    fig2 = px.bar(
        sleep_summary,
        x='SleepCluster',
        y=['TotalHoursAsleep', 'TotalHoursInBed', 'AverageSleepDuration'],
        title='Average Sleep Metrics by Cluster',
        labels={'value': 'Average (Hours)', 'variable': 'Metric'},
        barmode='group',
        text_auto=True,
    )
    fig2.update_layout(xaxis_title='Sleep Cluster', yaxis_title='Average Hours',
                       legend_title='Metric', width=1000, height=600 )
    fig2_json = pio.to_json(fig2)

    return jsonify({"fig1": fig1_json, "fig2": fig2_json})
```

Listing 2: sleepPatterns.py

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, r2_score
import plotly.graph_objects as go
import json
from flask import Blueprint, jsonify, request
import plotly.io as pio

bp = Blueprint('heartrateFluctuations', __name__)

@bp.route('/heartratefluctuations', methods=['GET'])
def display_hrfluctuations():

    # Extract parameters
    user_id = int(request.args.get('user_id'))

    # Load data
    df_heartrate = pd.read_csv('dataset/new-data/heartrate_seconds.csv', parse_dates=['Time'])

    # Filter data for the user
    # user_id = 2022484408, 4388161847
    user_data = df_heartrate[df_heartrate['Id'] == user_id]

    # Resample to hourly averages
    user_data.set_index('Time', inplace=True)
    user_data = user_data.between_time('07:00', '20:00')
    hourly_data = user_data['Value'].resample('1h').mean()
    print(hourly_data.head(10))

    # Interpolate missing values
    hourly_data = hourly_data.interpolate()

```

```python
    # Create lag features (using the last few hours to predict the next hour)
    lag_features = pd.concat([hourly_data.shift(i) for i in range(1, 4)], axis=1)
    lag_features.columns = ['Lag_1', 'Lag_2', 'Lag_3']

    # Add the target variable
    data = pd.concat([lag_features, hourly_data], axis=1)
    data.columns = ['Lag_1', 'Lag_2', 'Lag_3', 'Target']

    # Drop rows with NaN values (from lagging)
    data = data.dropna()

    # Train-test split
    X = data[['Lag_1', 'Lag_2', 'Lag_3']]
    y = data['Target']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    # Train a simple linear regression model
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Predictions
    y_pred = model.predict(X_test)

    # Evaluate the model
    rmse = mean_squared_error(y_test, y_pred)
    print(f'Root Mean Squared Error (RMSE): {rmse}')

    # Calculate MAPE
    mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100
    accuracy_lr = 100 - mape
    print(f"Accuracy: {accuracy_lr:.2f}%")

    # Interactive Plot with Plotly for Linear Regression
    fig_lr = go.Figure()
    fig_lr.add_trace(go.Scatter(y=y_test.values, mode='lines+markers', name='Actual Heart Rate', marker=dict(color='blue')))
    fig_lr.add_trace(go.Scatter(y=y_pred, mode='lines+markers', name='Predicted Heart Rate (Linear Regression)', marker=dict(color='red')))

    fig_lr.update_layout(
        title="Actual vs Predicted Heart Rate (Linear Regression)",
        xaxis_title="Time (Index)",
        yaxis_title="Heart Rate (bpm)",
        legend=dict(x=0.02, y=0.95),
        template="plotly_white",
        width=1100,
        height=600
    )
    fig_lr_json = pio.to_json(fig_lr)


    # Initialize Random Forest Regressor
    rf_model = RandomForestRegressor(n_estimators=100, random_state=42)

    # Train the model
    rf_model.fit(X_train, y_train)

    # Make predictions
    y_pred_rf = rf_model.predict(X_test)

    # Evaluate the model
    mae_rf = mean_absolute_error(y_test, y_pred_rf)
    r2_rf = r2_score(y_test, y_pred_rf)

    # Calculate MAPE
    mape_rf = np.mean(np.abs((y_test - y_pred_rf) / y_test)) * 100
    accuracy_rf = 100 - mape_rf

    # Print results
    print(f"Random Forest MAE: {mae_rf:.2f}")
    print(f"Random Forest Accuracy: {accuracy_rf:.2f}%")

    # Interactive Plot with Plotly for Random Forest
    fig_rf = go.Figure()
    fig_rf.add_trace(go.Scatter(y=y_test.values, mode='lines+markers', name='Actual Heart Rate', marker=dict(color='blue')))
    fig_rf.add_trace(go.Scatter(y=y_pred_rf, mode='lines+markers', name='Predicted Heart Rate (Random Forest)', marker=dict(color='green')))

    fig_rf.update_layout(
        title="Actual vs Predicted Heart Rate (Random Forest)",
        xaxis_title="Time (Index)",
        yaxis_title="Heart Rate (bpm)",
        legend=dict(x=0.02, y=0.95),
        template="plotly_white",
        width=1100,
        height=600
    )
    fig_rf_json = pio.to_json(fig_rf)

    # Compare with the previous model
    print("\nComparison:")
    print(f"Previous Model Accuracy: {accuracy_lr:.2f}%")
    print(f"Random Forest Accuracy: {accuracy_rf:.2f}%")

    return jsonify({
        "fig_lr": fig_lr_json,
        "fig_rf": fig_rf_json,
        "accuracy_lr": round(accuracy_lr, 2),
        "accuracy_rf": round(accuracy_rf, 2),
        "average_heart_rate": round(hourly_data.mean()),
        "resting_heart_rate": round(hourly_data.nsmallest(int(len(hourly_data) * 0.1)).mean()),
        "peak_heart_rate": round(hourly_data.max()),
    })
```

Listing 3: heartrateFluctuations.py