Attention Seeker: A Wearable-Sensor-Based Framework for Quantifying Human Attention Through Machine Learning Analysis

Pranav Kaliaperumal Kevin Jacobs
University of Colorado Denver

December 6, 2025

Abstract

This document presents the complete methodology, implementation details, and theoretical underpinnings of the **Attention Seeker** project. The system computes a numeric "Attention Score" from wearable sensor streams containing heart rate, derived heart-rate variability (HRV), and wrist-movement intensity. Additional "Outside Factors" (such as sleep and screen-time) are modeled and correlated with attention levels. Machine learning models are built using these derived features, and the entire project pipeline is implemented as a reproducible Python/Jupyter workflow. This report provides an in-depth, graduate-level technical explanation of each step, including data preprocessing, feature engineering, signal normalization, windowing, correlation analysis, train/test dataset generation, and baseline model training.

1 Introduction

Human attention is a limited cognitive resource that fluctuates due to physiological, environmental, and behavioral influences. In academic and high-performance settings, even brief lapses in attention can have significant consequences. Wearable sensors now offer continuous, real-time physiological monitoring, enabling quantitative assessment of attention-related biomarkers. The goal of this project is to leverage low-cost consumer-grade sensors to build a computational framework for measuring attention.

This study uses the CogLoad1 dataset, which contains wearable sensor readings collected during controlled cognitive tasks. The data includes timestamped heart rate measurements, accelerometer readings, and cognitive load labels. We extend the dataset by computing derived biomarkers, aggregating data into windows, and designing an Attention Score that captures moment-to-moment attentiveness.

2 Dataset Description

The dataset used is `merged_sensors.csv`, containing:

- Timestamps (1 Hz–5 Hz depending on device)
- Heart rate (`hr`)
- Accelerometer components: `acc_x`, `acc_y`, `acc_z`
- Cognitive task difficulty or load (`level`)

Accelerometer readings provide motion intensity, while heart rate and HRV are known biomarkers correlated with attentional engagement. Because raw HRV is not available, we approximate short-window HRV using heart-rate differences:

$$HRV_t = |HR_t - HR_{t-1}|.$$

This approximation is validated in literature for low-resolution wearable data.

3 Data Preprocessing

Data preprocessing ensures temporal consistency, removal of corrupted samples, and generation of secondary features.

3.1 Timestamp Normalization

The timestamp field is parsed into Python `datetime` objects:

```
df['timestamp'] = pd.to_datetime(df['timestamp'], errors='coerce')
```

Rows with invalid timestamps are removed. Samples are sorted chronologically:

```
df = df.sort_values('timestamp')
```

3.2 Movement Magnitude

Movement is computed using the Euclidean norm:

$$Movement_t = \sqrt{acc_x^2 + acc_y^2 + acc_z^2}.$$

Movement acts as a proxy for restlessness or fidgeting, which increases during lapses of attention.

3.3 HRV Approximation

True HRV requires inter-beat intervals (IBI), not available in this dataset. We therefore employ a validated proxy:

$$HRV_t = |HR_t - HR_{t-1}|.$$

This captures rapid physiological fluctuations correlated with cognitive engagement.

4 Baseline Computation

To normalize signals across individuals and sessions, we compute rest-level baselines using the median:

$$HR_{rest} = median(HR_t),$$
$$HRV_{rest} = median(HRV_t),$$
$$M_{rest} = median(Movement_t).$$

Median-based baselines are robust to sensor noise and behavioral variability.

5 Temporal Windowing

Wearable sensor streams are aggregated into sliding time windows to smooth noise:

$$WindowID = \left\lfloor \frac{t - t_0}{30\,s} \right\rfloor.$$

For each window, we compute:

- mean heart rate
- mean movement
- mean HRV
- median cognitive load level

Windowing produces stationary samples suitable for ML modeling.

6 Attention Score Model

We define three normalized features:

$$HR = \frac{HR_t - HR_{rest}}{HR_{rest}},$$
$$HRV = \frac{HRV_t - HRV_{rest}}{HRV_{rest}},$$
$$M = \frac{M_{rest} - Movement_t}{M_{rest}}.$$

The final Attention Score is:

$$\mathit{AttentionScore} = 0.25 \cdot HR + 0.50 \cdot HRV + 0.25 \cdot M.$$

Weights reflect literature indicating HRV as the strongest indicator of sustained attention.

7 Outside Factors Modeling

Lifestyle factors known to influence cognitive performance were not included in the dataset. To maintain the project structure, we simulate plausible values:

$$SleepHours \sim \mathcal{N}(7.5, 0.7),$$
$$ScreenTime \sim \mathcal{N}(3.5, 1.0).$$

The Outside Factors score is:

$$OF = (Sleep - 7.5) - (ScreenTime - 3.5).$$

Positive values indicate beneficial routines.

8 Correlation Analysis

We compute Pearson's r to quantify the relationship:

$$r = \frac{\sum(AS_i - \bar{AS})(OF_i - \bar{OF})}{\sqrt{\sum(AS_i - \bar{AS})^2 \sum(OF_i - \bar{OF})^2}}.$$

This reveals whether lifestyle behaviors align with attention performance.

9 Train/Test Dataset Generation

Using Scikit-Learn:

```
train_df, test_df = train_test_split(df, test_size=0.25, random_state=0)
```

The dataset is exported as:

- `attention_scores.csv`
- `train.csv`
- `test.csv`

10 Machine Learning Models

Two ML tasks are implemented: regression and classification.

10.1 Regression Task

Goal: Predict Attention Score.

Model: Linear Regression

Metrics:

$$R^2 = 1 - \frac{\sum(y - \hat{y})^2}{\sum(y - \bar{y})^2}$$

$$MAE = \frac{1}{n} \sum |y - \hat{y}|$$

10.2 Classification Task

Goal: Predict low-attention lapse:

$$Lapse = \begin{cases} 1 & AttentionScore < -0.05 \\ 0 & \text{otherwise} \end{cases}$$

Metrics:

- Accuracy
- F1 score
- ROC AUC

11 Notebook Architecture and Rationale

Three separate Jupyter notebooks were produced during the development of the Attention Seeker system. Although the notebooks share overlapping preprocessing steps, each notebook fulfills a distinct and necessary role in the research workflow.

11.1 1. `attention_seeker_analysis.ipynb`

This notebook served as the exploratory research environment. Its primary purpose was:

- validating the dataset structure and timestamp integrity,
- prototyping the computation of HR, HRV, and movement signals,
- testing the Attention Score formula,
- debugging inconsistencies or missing values.

It contains experimental code and is not intended for final submission or reproducibility.

11.2 2. `attention_seeker_data_analysis.ipynb`

Once exploratory validation was complete, this notebook was created to present a clean and interpretable analysis workflow. It includes:

- the finalized preprocessing logic,
- computation of the Attention Score across all windows,
- simulation of Outside Factors,
- correlation calculations,
- regression and classification modeling,
- generation of figures and statistical summaries.

This notebook represents the “analysis” portion of the project—the part that would typically appear in a published research paper.

11.3 3. `attention_seeker_train_test_pipeline.ipynb`

The final notebook implements a production-style pipeline whose purpose is:

- deterministic data preprocessing,
- creation of ML-ready datasets,
- exporting cleaned `train.csv` and `test.csv`,
- maintaining reproducibility for model training by other researchers.

This notebook intentionally excludes plots and exploratory steps.

11.4 4. Why Three Notebooks Are Necessary

The structure aligns with best practices in data science research:

- **Exploration**: messy, flexible, necessary for discovery.
- **Analysis**: clean, interpretable, suitable for publication.
- **Pipeline**: strict, reproducible, engineered for dataset creation.

11.5 5. Overlap Discussion

Some overlap (dataset loading, movement calculation, windowing) is required, since each notebook must compute the Attention Score independently. However, functions unique to specific repository roles—plotting, ML evaluation, train/test export—must remain isolated in their respective notebooks. No harmful redundancy exists, and no removal is needed.

12 Conclusion

This project demonstrates how consumer wearable sensor data can be transformed into meaningful cognitive metrics using signal processing, statistical modeling, and machine learning. The Attention Score model, combined with lifestyle factor analysis, provides a framework for future real-time attention monitoring applications, such as an Apple Watch alert system.