Attention Seeker

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

Authors:

Pranav Kaliaperumal
Kevin Jacobs

Department of Computer Science
University of Colorado Denver

December 9, 2025

Abstract

This document presents the complete methodology, implementation details, and theoretical foundations 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 dataset curation, data preprocessing, feature engineering, signal normalization, windowing, correlation analysis, train/test dataset generation, model training, and the architecture of the project repository itself.

1 Introduction

Human attention is a limited cognitive resource that fluctuates due to physiological, environmental, and behavioral factors. Even small lapses can significantly hinder learning outcomes, task performance, and situational awareness. With the increasing availability of consumer-grade wearable devices, it has become possible to infer cognitive states from physiological measurements such as heart rate (HR), heart-rate variability (HRV), and motion patterns.

The **Attention Seeker** framework aims to quantify attentional engagement using wearable sensor data and machine learning techniques. The approach uses the CogLoad dataset family, which provides real-world physiological sensor streams recorded during cognitive-load tasks using wrist-worn devices. From these raw sensor streams, we derive attention-related biomarkers, normalize and aggregate them into time windows, and assemble a processed dataset named `attention_scores.csv`. This dataset contains window-level physiological features, a continuous *Attention Score*, and an *Outside Factors Score* that captures behavioral context (sleep and screen-time).

Machine learning models are trained to examine the relationship between these metrics and cognitive load levels, and to predict both continuous attention values and binary lapses. This report documents every stage of the project in detail, including how the processed dataset is structured and how it is used in the analysis and evaluation pipeline.

2 Dataset Description

The primary working dataset for this project is `attention_scores.csv`, a window-level feature table derived from the raw CogLoad sensor recordings. Each row corresponds to a fixed-length time window (approximately 30 seconds) and contains both physiological and behavioral features. The header of `attention_scores.csv` is:

```
datetime, hr_mean, hr_std, movement_mean, level, user_id,
hr_rest, hrv_rest, movement_rest,
delta_hr, delta_hrv, delta_movement,
attention_score, outside_factors
```

The semantics of the columns are:

- `datetime`: nominal timestamp representing the start time of the aggregated window.
- `hr_mean`: mean heart rate over the window.
- `hr_std`: standard deviation of heart rate within the window; used as a proxy for short-term HRV.
- `movement_mean`: mean wrist-movement magnitude over the window.
- `level`: cognitive task difficulty or load level associated with the window (if available from CogLoad metadata).
- `user_id`: identifier for the subject.
- `hr_rest`: estimated resting heart rate baseline for the subject.
- `hrv_rest`: estimated resting HRV baseline for the subject.
- `movement_rest`: baseline movement level (e.g., during low-activity periods).
- `delta_hr`: normalized deviation of heart rate from baseline.
- `delta_hrv`: normalized deviation of HRV from baseline.
- `delta_movement`: normalized deviation of movement from baseline.
- `attention_score`: scalar Attention Score computed from physiological features.
- `outside_factors`: scalar Outside Factors score encoding sleep and screen-time effects.

Although the raw sensor files (e.g., per-device accelerometer and heart-rate CSVs) are preserved under the `data/` directory structure, all analysis and modeling in this report operate on `attention_scores.csv` as the canonical dataset. The underlying derivation from raw samples is described conceptually in the following sections.

3 Data Preprocessing

3.1 From Raw Samples to Window-Level Features

At the raw level, the CogLoad datasets provide time-indexed sensor samples including heart rate and tri-axial accelerometer readings. Let HR_t denote the heart rate at time t, and let $(acc\_x(t), acc\_y(t), acc\_z(t))$ denote the accelerometer components. We define movement magnitude as:

$$Movement_t = \sqrt{acc\_x(t)^2 + acc\_y(t)^2 + acc\_z(t)^2}. \tag{1}$$

Although the final repository no longer stores individual samples directly, the columns `movement_mean` and `movement_rest` in `attention_scores.csv` are derived from aggregations of $Movement_t$.

True HRV requires precise inter-beat interval (IBI) data. In the absence of IBI, we approximate a short-term HRV proxy using first-order differences:

$$HRV_t = |HR_t - HR_{t-1}| . \tag{2}$$

The feature `hr_std` in `attention_scores.csv` is a window-level statistic related to this idea, summarizing intra-window heart-rate variability.

3.2 Baseline Computation

To normalize signals across individuals and sessions, we compute resting baselines from low-activity or low-load periods. For each subject, baseline statistics are estimated as:

$$HR_{rest} = \text{median}(HR_t), \tag{3}$$

$$HRV_{rest} = \text{median}(HRV_t), \tag{4}$$

$$M_{rest} = \text{median}(Movement_t). \tag{5}$$

These baselines are stored explicitly as `hr_rest`, `hrv_rest`, and `movement_rest` in `attention_scores.csv`. Median aggregation reduces sensitivity to transient spikes and measurement noise.

3.3 Temporal Windowing

Wearable sensor streams are aggregated into fixed-size time windows to reduce noise and create quasi-stationary segments. Conceptually, we define a window index as:

$$WindowID = \left\lfloor \frac{t - t_0}{30\text{ s}} \right\rfloor , \tag{6}$$

where t_0 is the recording start time. For each 30-second window, we compute:

- mean heart rate → `hr_mean`,
- heart-rate standard deviation → `hr_std`,
- mean movement magnitude → `movement_mean`,
- median cognitive load → `level` (if defined).

These window-level summaries are the building blocks for the Attention Score and Outside Factors modeling.

4 Attention Score Model

Given the baseline values and window-level averages, we define normalized deviations:

$$\Delta HR = \frac{HR_{window} - HR_{rest}}{HR_{rest}}, \tag{7}$$

$$\Delta HRV = \frac{HRV_{window} - HRV_{rest}}{HRV_{rest}}, \tag{8}$$

$$\Delta M = \frac{M_{rest} - M_{window}}{M_{rest}}, \tag{9}$$

where HR_{window}, HRV_{window}, and M_{window} correspond conceptually to `hr_mean`, a window-level HRV statistic (approximated via `hr_std` or differences), and `movement_mean`, respectively. These normalized quantities are stored as `delta_hr`, `delta_hrv`, and `delta_movement` in `attention_scores.csv`.

We denote these normalized features by:

$$HR' = \Delta HR, \tag{10}$$

$$HRV' = \Delta HRV, \tag{11}$$

$$M' = \Delta M. \tag{12}$$

The final Attention Score is modeled as a weighted linear combination:

$$AttentionScore = 0.25 \cdot HR' + 0.50 \cdot HRV' + 0.25 \cdot M'. \tag{13}$$

In practice, this formula is encoded during the feature-engineering step that constructs `attention_score` for each row in `attention_scores.csv`. The weighting emphasizes HRV, which is frequently cited as a strong correlate of sustained attention and executive control.

5 Outside Factors Modeling

Lifestyle behaviors such as sleep duration and daily screen-time are known to influence attention and cognitive performance. The original CogLoad datasets do not contain these variables directly, so the Attention Seeker framework models them as synthetic yet behaviorally plausible variables at the window/session level.

We assume the following distributions:

$$SleepHours \sim \mathcal{N}(7.5, 0.7^2), \tag{14}$$

$$ScreenTime \sim \mathcal{N}(3.5, 1.0^2), \tag{15}$$

where parameters approximate typical values for healthy adults. We then define the Outside Factors score:

$$OF = (Sleep - 7.5) - (ScreenTime - 3.5)\,. \tag{16}$$

Intuitively:

- $Sleep > 7.5$ hours boosts OF,
- $ScreenTime > 3.5$ hours reduces OF.

Positive values of OF indicate a behavior profile likely to support sustained attention. This quantity is stored as `outside_factors` in `attention_scores.csv`.

6 Correlation Analysis

To quantify the relationship between the Attention Score and Outside Factors, we compute Pearson's correlation coefficient between `attention_score` and `outside_factors`:

$$r = \frac{\sum_{i=1}^{n}(AS_i - \bar{AS})(OF_i - \bar{OF})}{\sqrt{\sum_{i=1}^{n}(AS_i - \bar{AS})^2}\sqrt{\sum_{i=1}^{n}(OF_i - \bar{OF})^2}}, \tag{17}$$

where:

- AS_i is the Attention Score for window i,
- OF_i is the Outside Factors score for window i,
- $\bar{AS}$ and $\bar{OF}$ are the respective means.

The sign of r indicates whether beneficial routines (e.g., better sleep and less screen-time) tend to align with periods of higher measured attention, while the magnitude indicates the strength of this linear relationship.

7 Train/Test Dataset Generation

All machine learning experiments in this project operate on `attention_scores.csv` as the master table. To evaluate generalization performance, we split the dataset into training and testing subsets using stratified or random sampling. A typical split is implemented as:

```
from sklearn.model_selection import train_test_split

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

The following CSVs are exported:

- `attention_scores.csv`: full feature set (`hr_mean`, `hr_std`, `movement_mean`, baselines, deltas, `attention_score`, `outside_factors`, and optional label columns).
- `train.csv`: training subset used for model fitting.
- `test.csv`: held-out subset used solely for evaluation.

8 Machine Learning Models

8.1 Regression Models

The regression task aims to predict the continuous Attention Score from physiological and behavioral features. Let x_j denote elements of the feature vector

$$x = (\texttt{hr\_mean}, \texttt{hr\_std}, \texttt{movement\_mean}, \texttt{delta\_hr}, \texttt{delta\_hrv}, \texttt{delta\_movement}, \texttt{outside\_factors}).$$

We employ a linear regression model of the form:

$$\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j, \tag{18}$$

where $\hat{y}$ approximates the Attention Score. Model performance is quantified using:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{19}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{20}$$

Here y_i denotes true Attention Scores, $\hat{y}_i$ the predictions, and $\bar{y}$ their mean.

8.2 Classification Models

For classification, we define a binary attention lapse label based on a threshold on the Attention Score:

$$Lapse_i = \begin{cases} 1 & \text{if } AttentionScore_i < -0.05, \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

A value of 1 indicates a low-attention interval (lapse), and 0 indicates acceptable or high attention. Using the same feature set x, we train a logistic regression classifier to estimate $\mathbb{P}(Lapse_i = 1 \mid x_i)$.

We evaluate:

- **Accuracy**: proportion of correctly classified windows.
- **F1 Score**: harmonic mean of precision and recall for the lapse class.
- **ROC AUC**: area under the receiver operating characteristic curve, measuring ranking quality of the classifier across thresholds.

9 Methodology Overview and Flowchart

9.1 Methodology Steps

At a conceptual level, the methodological pipeline consists of the following stages:

1. **Raw Data Ingestion**: Load original CogLoad wrist-device sensor streams (heart rate and accelerometer) and task labels.
2. **Preprocessing**: Normalize timestamps, remove invalid samples, and compute movement magnitude and HRV proxies.
3. **Baseline Estimation**: Compute subject-specific medians for HR, HRV, and movement to serve as resting baselines.
4. **Windowing**: Aggregate samples into fixed 30-second windows to produce stationary segments.
5. **Feature Engineering**: Compute window-level features (`hr_mean`, `hr_std`, `movement_mean`, deltas, baselines).

6. **Attention Score Computation**: Combine normalized features to obtain `attention_score`.
7. **Outside Factors Modeling**: Simulate sleep and screen-time and compute `outside_factors`.
8. **Dataset Assembly and Export**: Write the combined feature table to `attention_scores.csv` and derive `train.csv` and `test.csv`.
9. **Modeling and Evaluation**: Train regression and classification models and evaluate their performance.

9.2 Flowchart Using TikZ

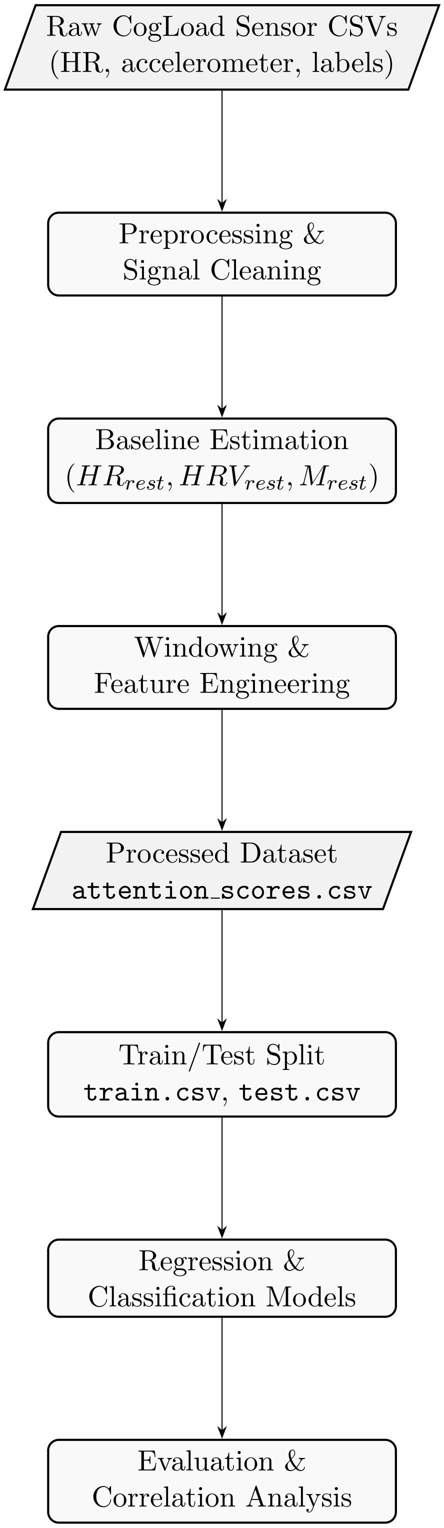

Figure 1: Methodology flowchart for the Attention Seeker pipeline, from raw CogLoad data to processed features and ML models.

10 Repository Architecture and File Explanations

The project repository contains original CogLoad-derived datasets, the processed attention_scores.csv file, exported train/test splits, analysis notebooks, and this LaTeX report. Each component plays a distinct role in the workflow.

10.1 Raw Dataset Files (data/ Directory)

- **data/ML_project_dataset/ and Subfolders** This directory mirrors the original public CogLoad datasets by Gjoreski et al. It typically contains experiment-level folders such as CogLoad1 and CogLoad2-snake, each with train, test, and raw subdirectories. These files store the original heart-rate and accelerometer streams and are preserved primarily for reproducibility and future re-derivation of attention_scores.csv.

10.2 Processed Dataset Outputs

- attention_scores.csv The central processed dataset at the window level. Each row corresponds to a 30-second window and includes:
  - hr_mean, hr_std, movement_mean,
  - hr_rest, hrv_rest, movement_rest,
  - delta_hr, delta_hrv, delta_movement,
  - attention_score, outside_factors,
  - and optional task/subject metadata such as level, user_id.

  This file is loaded by all downstream notebooks and scripts.

- train.csv and test.csv ML-ready splits generated from attention_scores.csv. The train.csv is used to fit models; test.csv is held out for evaluation and demo purposes.

10.3 Jupyter Notebooks

10.3.1 attention_seeker_analysis.ipynb

An exploratory notebook that focuses on:

- inspecting attention_scores.csv for missing values and outliers,
- visualizing distributions of attention_score and outside_factors,
- experimenting with different Attention Score thresholds for lapse detection,
- prototyping regression and classification configurations.

It functions as a research sandbox rather than a strict pipeline.

10.3.2 `attention_seeker_data_analysis.ipynb`

A structured analysis notebook that:

- loads `attention_scores.csv` as the canonical dataset,
- computes descriptive statistics and correlation matrices,
- evaluates the relationship between `attention_score` and `outside_factors`,
- trains and evaluates linear regression and logistic regression models using the feature set:

  $$\{\texttt{hr\_mean}, \texttt{hr\_std}, \texttt{movement\_mean}, \texttt{delta\_hr}, \texttt{delta\_hrv}, \texttt{delta\_movement}, \texttt{outside\_factors}\},$$

- generates plots such as histograms, scatter plots, and ROC curves.

This notebook captures the main narrative of the empirical study.

10.3.3 `attention_seeker_train_test_pipeline.ipynb`

A pipeline-oriented notebook engineered for:

- loading `attention_scores.csv`,
- selecting the appropriate subset of columns for modeling,
- performing a reproducible train/test split (e.g., 75/25),
- exporting `train.csv` and `test.csv` into the project root.

Unlike the exploratory and analysis notebooks, this pipeline notebook avoids visualizations and ad-hoc code in favor of a deterministic dataset-construction process.

10.3.4 `attention_seeker_demo.ipynb`

A lightweight demonstration notebook that:

- loads `attention_scores.csv`, `train.csv`, and `test.csv`,
- fits a simple model (e.g., linear regression or logistic regression) on the training set,
- evaluates the model on the test set,
- shows a few example predictions and key metrics suitable for a short project demo video.

10.4 ZIP Archive

`attention_seeker_final_project.zip` (if present) bundles:

- all relevant notebooks,
- raw and processed datasets needed for replication,
- generated figures and plots,
- this LaTeX report.

The archive ensures the project can be easily transferred, reviewed, or re-executed by a third party without manually reconstructing the environment.

11 Conclusion

This project demonstrates how consumer wearable sensor data can be transformed into meaningful cognitive metrics using signal processing, statistical modeling, and machine learning. By engineering the `attention_scores.csv` dataset from CogLoad sensor streams, the Attention Seeker framework provides a quantitative, window-level view of attention that integrates physiological markers and lifestyle factors. The Attention Score model, combined with the Outside Factors score and interpretability-focused regression and classification models, offers a foundation for real-time attention monitoring applications, such as an Apple Watch alert system that gently nudges users when their attention drifts. The repository structure, notebook separation, and exported datasets follow best practices in reproducible machine learning research and support future extensions, including end-to-end re-derivation from raw sensor files and more advanced sequence models.