Configuration Manual (Version 4.3.1)

N-of-1 Study – ADHD + Bipolar Disorder

Practicum Part 2 – MSc in Artificial Intelligence for Business

Rodrigo Marques Teixeira
National College of Ireland
Supervisor: Dr. Agatha Mattos

December 2025

Contents

1 Repository and Code Access

This manual documents the configuration and reproducibility details for Practicum Part 2. It covers ETL, modeling, explainability, and ethical governance for the "A Deterministic N-of-1 Pipeline for Multimodal Digital Phenotyping in ADHD and Bipolar Disorder".

1.1 System Architecture

Components: data sources (Apple Health, Amazfit GTR2, Amazfit GTR4, Apple Auto Export app, SOM is used instead of EMA), ETL pipeline, modeling notebooks, drift analysis and SHAP explainability. The overall system architecture is shown in Figure 1.

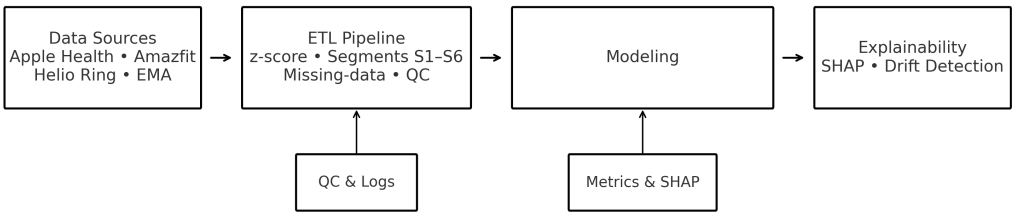

Figure 1 – System architecture overview (simplified modeling block)

Figure 1: System architecture overview.

1.2 Environment Requirements

Python 3.11+ with the dependencies listed in `requirements.txt`. Details for source code access and execution are provided in Section 5.

1.3 Data Sources and ETL Overview

The data used in this study originate from wearable and smartphone-based sensors, collected continuously over multiple months as part of an N-of-1 longitudinal experiment on ADHD and Bipolar Disorder comorbidity. Primary data sources include Apple Health (heart rate, heart rate variability, sleep stages, step count), Amazfit devices (heart rate and HRV), complemented by Apple "State of Mind (SoM)" self-reports.

All raw data are exported in CSV format and processed locally through a reproducible Python pipeline (`etl_pipeline.py`). The ETL system is responsible for daily feature generation, data normalization, segmentation, and quality control. The overall flow is shown in Figure 2.

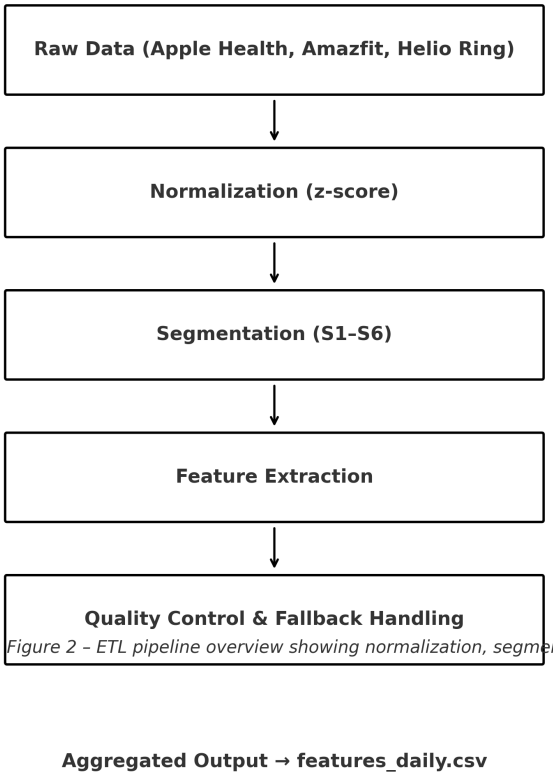

Figure 2 – ETL pipeline overview showing normalization, segmentation, feature extraction and QC

Figure 2: ETL pipeline overview showing normalization, segmentation, feature extraction and quality control.

1.4 Ethics and Governance

Long-term passive sensing raises important considerations around privacy, autonomy and data governance. Although the present study relied exclusively on the author's own data and used a fully local, reproducible processing pipeline, extending this approach to other participants would require formal ethics approval, informed consent procedures, anonymisation strategies and clear communication about data use and retention. The ability to run the pipeline locally without cloud dependence supports privacy-preserving research, but wider deployment must follow established clinical and ethical guidelines.

This self-tracking project did not require formal Research Ethics Committee (REC) approval under local guidelines, as the sole participant is also the investigator. Any future multi-participant extension would require full ethics review and informed consent.

2 Data Availability and Privacy

Raw health data cannot be distributed publicly due to privacy requirements.

For Evaluators with Raw Data Access:

Evaluators may request access to the Apple Health export ZIP and the Zepp ZIP archive through the principal investigator. These files are shared via OneDrive (NCI secure storage)

exclusively for academic evaluation. The expected directory layout for raw data is:

```
data/raw/
  P000001/apple/export/apple_health_export_20251210T060000Z.zip
  P000001/zepp/3088235680_1761192590962.zip
```

For External Researchers (Public Reproducibility):

To enable reproducibility without requiring access to private raw data, the repository includes a **public ETL snapshot** containing all processed daily features:

```
data/etl/P000001/2025-12-09/
  extracted/              # Daily CSVs (stage 1 output)
  joined/                 # Unified features (stage 2 output)
  qc/                     # Quality control reports
```

This snapshot allows full replication of ML6 and ML7 experiments (stages 3–9) without requiring raw health exports. The ETL snapshot contains:

- 2,868 days of processed data (2017-12-04 to 2025-12-07)
- 47 daily features (cardio, activity, sleep, screen time)
- PBSI behavioral annotations and 77 SoM-labeled days
- Quality control metrics
- Pipeline version 4.3.1 outputs

Note: All ML6/ML7 results in the paper use the full dataset of 2,868 days.

The snapshot does not contain identifying information and follows GDPR de-identification guidelines.

3 Snapshot Parameters

The final reproducible state of the study is defined by:

- **Participant**: `P000001`
- **Snapshot**: `2025-12-09`
- **Seed**: `42`
- **Pipeline**: `v4.3.1`

All reported metrics, figures and tables in Sections 4 and 5 refer exclusively to this snapshot.

4 Directory Structure

The deterministic directory layout is:

```
data/
  raw/                    # Raw exports (private)
  etl/P000001/2025-12-09/
    extracted/            # XML/CSV from Apple Health and Zepp
    joined/               # Daily unified features and labels
    ai/                   # ML6/ML7 outputs, SHAP, drift
    qc/                   # Automated QC reports (CSV + JSON)
```

Configuration files:

```
config/
  settings.yaml           # Main pipeline configuration
  participants.yaml       # Participant metadata
  label_rules.yaml        # PBSI thresholds and definitions
```

5 Canonical Entry Points

5.1 Full Pipeline (Stages 0–9, Requires Raw Data)

To execute the complete ETL pipeline from raw data:

```
python -m scripts.run_full_pipeline \
    --participant P000001 \
    --snapshot 2025-12-09 \
    --start-stage 0 \
    --end-stage 9
```

This command processes raw Apple Health export and Zepp ZIPs through all 10 stages.

5.2 Public Reproducibility (Stages 3–9, ETL Snapshot Only)

External researchers can reproduce stages 3–9 using the public ETL snapshot:

```
python -m scripts.run_full_pipeline \
    --participant P000001 \
    --snapshot 2025-12-09 \
    --start-from-etl \
    --end-stage 9
```

The `-start-from-etl` flag:

- Skips raw data extraction (stages 0–2)
- Loads pre-processed daily features from `data/etl/P000001/2025-12-09/`
- Executes labeling, segmentation, ML6, ML7, TFLite export and reporting
- Produces identical results (deterministic, seed=42)

This mode allows complete replication of all modeling results (Section 4 and 5) without requiring access to raw health data.

For detailed instructions, see `docs/REPRODUCING_WITH_ETL_SNAPSHOT.md`.

5.3 Individual Stage Execution

To run specific stages only:

```
# Only ML6 modeling (stage 6)
python -m scripts.run_full_pipeline \
    --participant P000001 \
    --snapshot 2025-12-09 \
    --start-from-etl \
    --start-stage 6 \
    --end-stage 6
```

6 Determinism and Reproducibility

Determinism is enforced through:

- fixed seeds across Python, NumPy and TensorFlow
- segment-wise z-score normalisation (no global leakage)
- calendar-faithful cross-validation
- explicit file paths
- stochastic imputation with fixed random seed (fully reproducible given same config)
- reproducible intermediate outputs saved at each stage
- Experiment Suite for controlled feature-set ablation (FS-A through FS-D)

Running the pipeline twice under identical configuration produces identical:

- unified features
- SoM and PBSI labels

- `model_selection.json` (Experiment Suite output)
- ML6-Extended/ML7-Extended matrices and leaderboards
- evaluation metrics
- drift statistics
- QC outputs

7 Automated QC (Optional but Recommended)

Before modelling, evaluators may run:

```
make qc-hr
make qc-steps
make qc-sleep
make qc-all
```

These routines validate HR, steps and sleep features through fabrication detection, range checks and cross-file consistency tests.

8 Minimal Reproduction Guide

8.1 For Evaluators with Raw Data Access

1. Clone the repository.
2. Place the raw data ZIPs in `data/raw/` (if available).
3. Confirm `config/settings.yaml` points to the 2025-12-09 snapshot.
4. Run the full pipeline (stages 0–9) using `scripts.run_full_pipeline`.
5. Inspect outputs under `data/etl/P000001/2025-12-09/`.
6. Compare ML6/ML7 results with Section 4 and 5.

8.2 For External Researchers (Public ETL Snapshot)

1. Clone the repository: `git clone <repo_url>`
2. Verify ETL snapshot exists: `ls data/etl/P000001/2025-12-09/`
3. Install dependencies: `pip install -r requirements/base.txt`
4. Run from ETL snapshot (stages 3–9):

```
python -m scripts.run_full_pipeline \
    --participant P000001 \
    --snapshot 2025-12-09 \
    --start-from-etl \
    --end-stage 9
```

5. Inspect outputs under `data/ai/P000001/2025-12-09/`.
6. Compare ML6/ML7 results with Section 4 and 5.

Both procedures reproduce the complete analysis underlying the study. The ETL-only mode (Subsection 8.2) produces identical results to the full pipeline (Subsection 8.1) for stages 3–9.