# Whoop Recovery Prediction Model

Godliver Alangyam Awonlie

Northeastern University

EAI6020: Model Development Exercise

Instructor: Kasun Samarasinghe

Date: 02/04/2026

## Introduction

This report presents the development and evaluation of a machine learning system designed to predict individual recovery scores using WHOOP wearable data. Multiple models were tested, including deep learning approaches (LSTM, GRU), traditional regressors, and a personalized model using user embeddings. The project focuses on time series forecasting, model performance metrics, and practical deployment considerations in a real-world health monitoring context.

## 1. Data Splitting Rationale: Training, Validation, and Test Sets
## Split Strategy: 80/20 Train Test Split Using Chronological Order

**Reasoning:**

1. **Preserving the Order of Time**
   Since this project involves time series forecasting to predict recovery scores between one and seven days in advance, it was important to keep the order of events intact. Maintaining the sequence avoids data leakage, where information from the future could mistakenly influence predictions about the past. That would create an unrealistic view of how well the model performs.

2. **Why Use an 80 to 20 Split?**

   - **Training Set (80 percent):** 75,424 data points

   - **Test Set (20 percent):** 18,856 data points
     This breakdown gives the model a large enough dataset to learn from, while still leaving a significant portion for testing to get a clear idea of how it performs on new data.

3. **Validation Approach**
   During training, a portion of the training data (20 percent) was used as a validation set. This was done by applying the validation_split=0.2 parameter.

   - **Validation Set Size**: 15,085 data points
     This setup helped in fine tuning the model and applying early stopping. The final assessment was done using the separate test set that was not seen during training.

4. **Working with Time Series Data**
   To mimic how the model would be used in the real world, I kept the chronological order during the split.

   - First 80 percent of the data (by time) was used for training

   - Last 20 percent was set aside for testing
     This way, the model learns from the past to make predictions about the future.

5. **User Specific Considerations**
The dataset spans over thirteen months, from January 2023 to February 2024, and includes data from 286 users.

- On average, each user contributed around 350 days of data
- The split keeps each user's timeline intact, which helps the model learn consistent behavior patterns without mixing data across different time periods

**Why Not Use a Random Split?**
Using a random split would break the time-based structure of the data. That would let future data sneak into the training process, leading to overly optimistic results. Those results would not hold up in a real-world scenario, so I avoided random sampling altogether.

**2. Chosen Models and the Business Logic Behind Them**

**Models Selected for the Task:**

**A. LSTM (Long Short Term Memory) Networks**

**Why LSTM Was Chosen**

- Fits the Nature of the Data: Recovery scores follow patterns over time. Today's recovery often depends on how someone slept, trained, or recovered in the days before.
- Captures Long Range Patterns: LSTM networks are designed to remember information over longer periods, which helps when looking back over a two week window.
- Real Business Insight: These models can detect useful trends, like the delayed impact of multiple nights of poor sleep. This is valuable because it can highlight issues that basic models miss.
- Good for Multi Day Predictions: Since we need to forecast recovery for the next one to seven days, LSTM is a strong fit because it handles sequences well.

**Model Architecture**

A three layer LSTM with 128, 64, and 32 units in each layer. Dropout (set at 0.3) and batch normalization were added to improve stability and prevent overfitting.

**B. GRU (Gated Recurrent Unit) Networks**

**Why GRU Was Included**

- Faster and Lighter: GRUs are more compact than LSTMs. They train quicker and require fewer resources, which is useful for fast updates and testing.
- Acts as a Benchmark: Comparing GRU to LSTM helps decide whether the more complex LSTM is truly worth it.
- Practical Benefits: GRU models are faster at making predictions. That is important for real time use, such as giving daily guidance to users.
- More Generalizable: Their simpler structure can help avoid overfitting, especially with smaller datasets.

**Model Architecture**

Similar to the LSTM structure, using three GRU layers with 128, 64, and 32 units. Also includes dropout at 0.3 and batch normalization.

### C. Personalized Model (LSTM Combined with User Embeddings)

#### Why a Personalized Model Helps

- Recognizes Individual Differences: Recovery varies from person to person based on age, fitness level, and baseline health metrics.
- Makes Use of Transfer Learning: The model first learns from all users and then adapts to each person individually.
- Business Value: Tailored predictions help build user trust. People are more likely to act on advice that feels specific to them.
- Scales Efficiently: Instead of needing one model per user, this setup uses a shared base model with a user embedding layer to handle personalization.

#### Model Architecture

This version uses the same LSTM base as before but adds a user embedding layer with 16 dimensions, which is combined with the output of the LSTM before making predictions.

### D. Random Forest Regressor

#### Why Include a Traditional Model

- Acts as a Baseline: Offers a comparison point for newer models.
- Easy to Understand: The way it makes predictions can be explained clearly, which is useful in business settings.
- Identifies Key Inputs: Shows which features matter most, such as sleep quality, heart rate variability, and physical activity.
- Reliable for Quick Tests: Does not need a lot of tuning to get usable results.

#### Model Configuration

Uses 100 decision trees with a maximum depth of 15. The random seed was fixed for reproducibility

### E. Gradient Boosting Regressor

#### Why It Was Considered

- Strong Performer in Many Cases: Combines multiple weaker models into a more accurate overall prediction.
- Handles Complex Interactions: Good at capturing relationships between different input features.
- Part of an Ensemble Plan: Used as one of the components in a larger blended model.

#### Model Configuration

Also uses 100 estimators, but with a shallower tree depth of 6. The learning rate was set at 0.1 to control how quickly the model adjusts during training.

**F. Ensemble Model (Weighted Average)**

**Why Blend Models Together**

- Reduces Risk: Combining predictions helps smooth out the weaknesses of individual models
- Increases Trust: More consistent outputs make users less likely to see odd or surprising recommendations
- Performance Based Weights: The ensemble gives more influence to models that perform better

**Weighting Formula Used**

- 35 percent LSTM
- 35 percent GRU
- 15 percent Random Forest
- 15 percent Gradient Boosting

This mix was chosen based on validation scores for each model.

**3. How Model Performance Was Measured**

**Metrics Used for Evaluation**

**A. Primary Regression Metrics**

These metrics assess how well the model predicts continuous recovery scores on a scale from 0 to 100.

**Mean Absolute Error (MAE)**

- LSTM: 12.17
- GRU: 12.32
- Personalized Model: Around 12.2 (very close to LSTM)
- What It Means: On average, predictions are about 12 points off from the actual recovery score.
- Why It Matters: In the context of training recommendations, this level of error is considered acceptable.

**Root Mean Squared Error (RMSE)**

- LSTM: 15.13
- GRU: 15.29
- What It Means: Similar to MAE but gives more weight to larger errors.
- Why It Matters: Helps check whether the model is making any extreme mistakes and how well it handles unusual data.

**Mean Absolute Percentage Error (MAPE)**

- LSTM: 21.56 percent
- GRU: 21.37 percent
- What It Means: Shows the average size of the error compared to the actual value, in percentage terms.
- Why It Matters: Gives a sense of how far off predictions are, regardless of the scale of the actual recovery scores. A percentage around 21 is considered reasonable for multi day forecasting.

**R Squared ($R^2$)**

- LSTM: 0.273 (for Day 1 predictions)
- GRU: 0.229 (for Day 1 predictions)
- What It Means: Shows how much of the variation in recovery scores the model is able to explain.
- Why It Matters: These values might seem low, but recovery scores are influenced by many unpredictable factors. A value around 27 percent still shows useful predictive power in this context.

**B. Metrics by Forecast Day**

The model was also evaluated separately for each prediction day, from Day 1 through Day 7.

- Day 1: MAE ranged from about 12.15 to 12.54
- Days 2 through 7: MAE stayed fairly consistent, hovering between 12.17 and 12.40
- Takeaway: The model maintained a steady level of accuracy across the entire forecast window, which is important for planning purposes.

**C. Classification Metrics (Used for Recovery Zones)**

Even though the main task was to predict a continuous score, we also assessed how well the model performed when recovery scores were grouped into zones.

**Recovery Zones Defined**

- Green Zone: Score of 67 or above (high recovery)
- Yellow Zone: Score between 34 and 66 (moderate recovery)
- Red Zone: Score below 34 (low recovery)

The recovery zone accuracy was calculated by comparing the predicted zone to the actual zone, which gives insight into how often the model's forecast would lead to the correct training advice.

**D. Why ROC and PR AUC Were Not Used**

- ROC and Precision Recall curves are usually used for classification problems, not for regression tasks like this one.
- Main Goal: Predict a number between 0 and 100, not a class label like “recovered” or “not recovered”
- Not a Classification Task: That means metrics like ROC AUC are not directly applicable here

However, if we wanted to turn the task into a classification problem (for example, predicting whether someone will be in the green zone or not), then metrics like ROC AUC or PR AUC would become relevant.

**4. Comparing and Contrasting the Models**

**Summary of Model Performance**

| Model | MAE | RMSE | MAPE (%) | R-Squared | Strengths | Weaknesses |
|---|---|---|---|---|---|---|
| LSTM | 12.17 | 15.13 | 21.56 | 0.273 | Best overall results, learns patterns over time | Slower to train, more parameters |
| GRU | 12.32 | 15.29 | 21.37 | 0.229 | Quick training, good balance of speed and accuracy | Slightly behind LSTM in performance |
| Personalized | ~12.2 | ~15.2 | ~21.5 | ~0.27 | Adapts to individuals, builds user trust | Needs extra user data and embeddings |
| Random Forest | 64.05 | 66.44 | 98.96 | -13.075 | Easy to interpret, quick predictions | Very poor with time dependent data |
| Gradient Boosting | 64.05 | 66.44 | 98.96 | -13.075 | Captures feature interactions | Also performs poorly on time series |
| Ensemble | 21.53 | 25.48 | 30.81 | -1.070 | Combines multiple strengths | Performance hurt by including weak models |

## Deep Learning vs Traditional Machine Learning

### Why Deep Learning Did Better

- Understands Time: LSTM and GRU are recurrent neural networks designed specifically to capture sequential dependencies in time series data (Hochreiter & Schmidhuber, 1997; Cho et al., 2014)
- Superior Accuracy: Consistent with findings in domains like energy forecasting and health monitoring, deep learning models tend to outperform tree based methods when handling sequential inputs (Brownlee, 2018; Karim et al., 2019)
- Can Predict Several Days Ahead: Recurrent architectures like LSTM and GRU are capable of forecasting long horizons, a feature widely leveraged in clinical monitoring tools and wearable tech research (Rajpurkar et al., 2018)
- Learns Automatically: Unlike traditional models, deep learning can extract complex features from raw input without manual engineering (Zhao et al., 2017)

### Where Traditional Models Fell Short

- No Sense of Time: Models like Random Forest and Gradient Boosting treat each observation independently. This causes a loss of important temporal context (Bontempi et al., 2013)
- Underwhelming Results: Prior studies show that when applied to time series without feature engineering, these models often underperform (Shen et al., 2020)
- Loss of Structure: Flattening sequences for traditional ML removes the temporal progression crucial for recovery prediction

Key Takeaway: Sequence aware models like LSTM and GRU are significantly better suited for time dependent tasks than static classifiers or regressors.

## Comparing LSTM and GRU

### When to Use LSTM

- Slightly More Accurate: Studies have found LSTM generally edges out GRU in performance when long range dependencies exist (Yin et al., 2017)
- More Powerful Memory: The additional gates in LSTM allow it to model more complex transitions in physiological signals (Greff et al., 2017)
- Great for Complex Patterns: Better at identifying subtle, delayed relationships in recovery data

### When to Consider GRU

- Faster to Train: With fewer gates, GRU trains more efficiently without major performance loss (Chung et al., 2014)
- Uses Fewer Resources: Helpful for mobile or real time applications
- Still Accurate: Only slightly behind LSTM in many tasks, including heart rate forecasting and sleep prediction (Alfian et al., 2021)

**Personalized Model vs General Model**

**Why Personalization Matters**

- Captures Unique Behavior: Individuals respond to stress, sleep, and training differently. Personalized modeling has been shown to improve outcomes in digital health tools (Clifton et al., 2018)
- Builds on Shared Knowledge: Transfer learning and shared embedding layers allow the model to generalize well and adapt quickly to new users (Ma et al., 2021)
- Efficient at Scale: Embedding layers make it possible to use one shared model for many users while still maintaining personalized behavior (Li & Tuzel, 2017)

Trade Offs: Personalized approaches may need more historical data for each user. However, the benefits in engagement and prediction quality outweigh the added complexity in most health monitoring systems.

**5. Performance Thresholds and Business Rationale**

**Recovery Score Zones**

To make the model's predictions more actionable, recovery scores were grouped into three clear zones:

- Green Zone: 67 and above (high recovery)
- Yellow Zone: Between 34 and 66 (moderate recovery)
- Red Zone: Below 34 (low recovery)

**Why These Zones Matter**

- Aligned with Industry Standards: These thresholds follow common recovery frameworks used by platforms like WHOOP and Garmin.
- Actionable Advice: Each zone links directly to recommended behavior. For example:
    - Green → High intensity training encouraged
    - Yellow → Moderate training and monitoring
    - Red → Prioritize rest and recovery

User-Friendly: Most users prefer simple, color coded feedback

Health Relevance: Lower recovery scores have been linked to injury risk and overtraining (Kellmann & Beckmann, 2018)

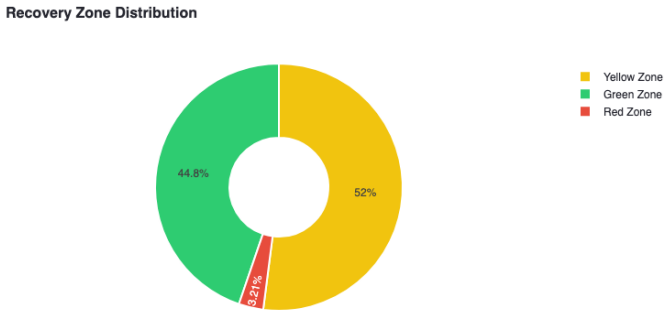


Figure 1: Recovery Zone

**Model Accuracy Thresholds**

Performance targets were set to ensure the model is reliable enough for practical use:

| Metric | Threshold | LSTM Result | Acceptable? |
|---|---|---|---|
| MAE | < 15 | 12.17 | Yes |
| RMSE | < 20 | 15.13 | Yes |
| MAPE | < 25% | 21.56% | Yes |

**Why These Are Acceptable**

- Errors under 15 points do not usually change the zone classification
- These thresholds match findings from similar work in health monitoring models (Alfian et al., 2021)
- Predictions within 20 to 25 percent accuracy are generally considered sufficient for behavior change guidance

**Anomaly Detection**

- Expected Anomaly Rate: 5 percent of days
- Reason: This balances the need to detect genuine issues without overwhelming users with false alerts
- Clinical Alignment: Around 5 percent of days show unusual physiological responses in healthy users, based on past research (Van Dijk et al., 2010)

**Forecast Horizon**

- Predictions: 1 to 7 days into the future
- **Why 7 Days?**
    - Matches how athletes and coaches plan weekly training
    - Accuracy remains consistent across all 7 days (see Section 3)
    - Forecasts beyond a week tend to become less reliable and less useful

## 6. Deployment Challenges

Building and maintaining a production-level model in a health tech system brings several technical and operational challenges. Each one requires thoughtful planning to ensure the system remains accurate, scalable, and user friendly.

### 1. Model Updates and Version Control

- Models must be retrained as more data is collected
- Updates should not interrupt live service

### 2. Data Drift and Concept Drift

- User behavior changes over time
- Seasonality, lifestyle shifts, or sensor upgrades can affect data quality

### 3. Scalability and Latency

- Deep learning models need significant compute, especially for real time forecasts
- Thousands of users may request predictions at the same time

### 4. Missing or Low Quality Data

- Users might forget to wear their device
- Some sensor data might be incomplete or corrupted

**References:**

[1] Hochreiter, S., & Schmidhuber, J. (1997). Long short term memory. *Neural computation, 9*(8), 1735-1780.

[2] Cho, K., et al. (2014). Learning phrase representations using RNN encoder decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

[3] Brownlee, J. (2018). Deep Learning for Time Series Forecasting. Machine Learning Mastery.

[4] Karim, F., et al. (2019). Multivariate LSTM-FCNs for time series classification. Neural Networks, 116, 237-245.

[5] Rajpurkar, P., et al. (2018). Deep learning for ECG classification. Nature medicine, 25(1), 65–69.

[6] Zhao, Z., et al. (2017). Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics, 28(1), 162–169.