Deep Sleep Prediction Using Machine Learning

Group 1

Noah Wolters, Caroline Hesse, and Kieran Keesmaat
2848625 2690272 2843427

Vrije Universiteit, Amsterdam 1081 HJ, Netherlands

1 Research Question

Deep sleep is essential for physical recovery, including muscle repair, immune function, and growth-hormone release, and for cognitive health by consolidating memory and regulating emotions [9]. Predicting how much deep sleep a user will get each night can, therefore, let them adjust daily activities and evening routines for better health. Thus, the goal of this project is to predict the amount of deep sleep (in minutes) for the upcoming night based on a range of physiological, behavioral and environmental variables. Physiological and behavioral data were gathered from a Garmin smartwatch between January 1st and May 20th, 2025, and included variables related to sleep, heart rate and physical activity. Additionally, environmental data including temperature was obtained from online sources [12] [11] [13]. The full list of attributes are outlined further in the following section.

2 Data summary

An overview of the included numerical variables can be found in Table 1, where the descriptive statistics of each is given. While most variables are self-explanatory, a brief outline of the less obvious ones are given in Table 2.

Four other attributes were also included, namely: Bed-time (what time the user fell asleep), Wake-time (what time the user woke up), Sunset-time and Sunrise-time. All data are aggregated at a daily timescale as this aligns with predicting the total minutes of deep sleep per night. The decision was made to direct the assignment in this direction as one of the group members had access to a wide range of sensory data from their Garmin smartwatch. This offered the chance to explore more meaningful and unique avenues as compared to the limited data able to be obtained in one week using phone sensors. Furthermore, this project may offer real actionable insights by identifying variables that are positively or negatively associated with deep sleep. This makes the project much more exciting, meaningful and practical.

As can be seen from Table 1, 3% of data is missing from *Resting Heart Rate (RHR)*, *HRV-status* and variables related to sleep. Furthermore, *Weight (kgs)*, *Run (kms)* and *Ride (kms)* are missing 55%, 82% and 54% respectively.

Table 1: Descriptive statistics for quantified self dataset

Variable	**Count**	**Missing**	**Mean**	**Std**	**Min**	**Max**
Resting heart rate	136	2.86%	58.43	3.05	52	64
Breaths per minute	136	2.86%	12.83	0.64	11	15
HRV-status	136	2.86%	34.22	2.59	28	40
Deep sleep (mins)	136	2.86%	64.55	18.96	20	111
Restless moments	136	2.86%	46.36	9.99	15	72
Sleep duration (mins)	136	2.86%	481.39	99.11	186	883
Steps	140	0.00%	10284	5482.12	2447	45992
Calories burned	140	0.00%	3269.29	592.71	2343	6087
Weight (kgs)	63	55.00%	80.8	2.26	75.5	84
Max HR	140	0.00%	137.88	18.47	101	188
Run (kms)	25	82.14%	5.91	2.32	2.02	10.01
Ride (kms)	65	53.57%	16.78	12.30	3.64	74.13
Intense exercise (mins)	140	0.00%	80.65	73.73	0	339
AvgTemp_C	140	0.00%	10.96	4.01	-1.5	19.5
MinTemp_C	140	0.00%	6.23	4.07	-5	13
MaxTemp_C	140	0.00%	15.50	4.53	0	27
Humidity_Max	140	0.00%	91.79	9.66	60	100
Humidity_Avg	140	0.00%	71.82	11.63	41.3	92
Humidity_Min	140	0.00%	50.99	13.55	24	82
WindSpeed_Max	140	0.00%	24.43	6.91	9	44
WindSpeed_Avg	140	0.00%	15.15	5.40	4	32.3
WindSpeed_Min	140	0.00%	6.16	4.86	0	28
Pressure_Max	140	0.00%	1020.26	7.10	1002.5	1036.5
Pressure_Avg	140	0.00%	1017.64	7.53	998.9	1034.9
Pressure_Min	140	0.00%	1015.39	7.84	996.5	1031.5
Precipitation_mm	140	0.00%	0.33	1.00	0	5.5

Further exploration of the data revealed normal distribution for all variables apart from *Precipitation_mm*, *WindSpeed_Min* and *Intense exercise (mins)* which showed right hand skews, and *Humidity_Max* which showed a left hand skew. Figures 1 and 2 show an example of one of the normally distributed variables (*sleep duration (mins)*) and one that was skewed (*Intense exercise (mins)*).

Lastly, associations between *Deep Sleep* and other variables were investigated and the strongest 14 associations were plotted in Figure 3. A number of weather variables show negative associations with *Deep Sleep*, indicating that the weather conditions might be interesting to look at in feature engineering. Still, since these are merely associations no causal relationships can be established.

Table 2: Definitions of daily activity and sleep variables

Variable Name	Explanation
Breaths per minute	Number of breaths taken per minute during sleep.
HRV-status	Average heart-rate variability status during sleep.
Restless moments	Number of periods of movement detected during sleep (e.g. tossing and turning).
Run (kms)	Total distance (in kilometers) run that day, tracked via GPS.
Ride (kms)	Total distance (in kilometers) ridden (e.g. cycling) that day, tracked via GPS.
Minutes of intense exercise	Minutes counted as "active" once one's heart rate exceeds a threshold relative to resting heart rate (or steps/minute threshold).

Fig. 1: Distribution of intense exercise

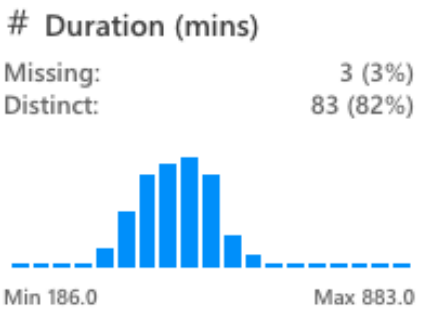


Fig. 2: Distribution of Sleep duration

3 Data cleaning

3.1 Outlier Removal

Initial Observations To analyze outliers, we began by examining the distribution histograms and time-series plots of all numerical variables. While inspecting the outcome variable *deep sleep*, we observed a notably high value in its distribution. However, upon reviewing the time-series plots for *deep sleep*, *REM sleep*, and *total sleep duration*, we noticed that all three variables were elevated on the same night. This pattern suggested that the value was not an outlier specific to deep sleep, but rather a result of the subject obtaining an unusually large amount of sleep that night. From this, we decided that a more comprehensive method for identifying genuine outliers was necessary to avoid misclassifying contextually valid data points.

Gaussian Mixture Model for Outlier Detection We, therefore, employed a Gaussian Mixture Model (GMM)-based approach. The GMM is a probabilistic

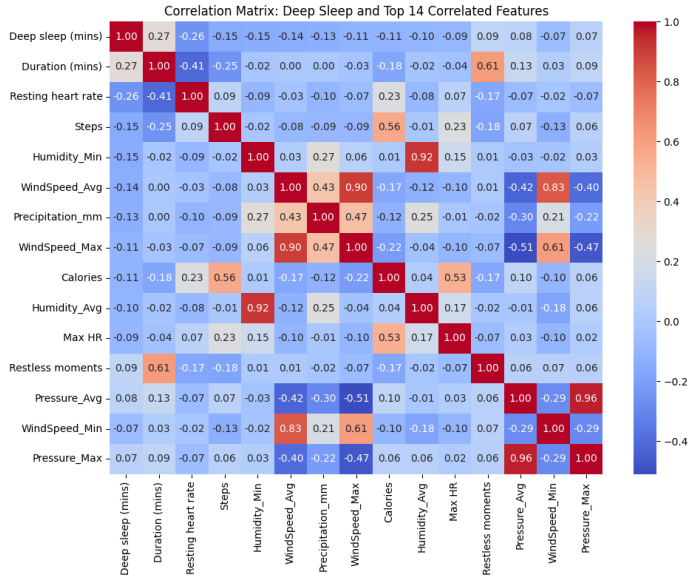


Fig. 3: Correlation Matrix: Deep Sleep and Top 14 Correlated Variables

model that assumes the data is generated from a mixture of several Gaussian distributions. This modeling choice was particularly motivated by the characteristics of behavioral sensor data: it is plausible that not all observations stem from a single underlying distribution. Instead, the data may systematically differ across distinct behavioral modes of the subject, such as active versus inactive states, like studying versus exercising days.

For each variable in our dataset, we first fitted a two-component Gaussian Mixture Model (GMM) to all valid (i.e., non-missing) observations. We then evaluated each data point by computing its log-likelihood under the fitted model. To identify outliers in an adaptive manner, we calculated the mean (μ) and standard deviation (σ) of these log-likelihood scores and defined an outlier threshold as $\mu - 3\sigma$ a heuristic drawn from prior applications of GMMs in the literature [7]. Any observation whose log-likelihood fell below this threshold was marked as an outlier.

The GMM-based method flagged 47 values across 20 variables as potential outliers. Each flagged observation was manually reviewed. While some values did appear extreme, such as a resting heart rate of 49 bpm or a step count of 45,992, they were not implausible. We were cautious not to remove or replace these values prematurely, as extreme observations can carry meaningful and interesting information, particularly in behavioral data where variability is expected. Since no values were found to be definitively erroneous, we chose to retain all flagged observations. To preserve the insight of this step, we added an indicator column marking these values. The goal was to allow for flexibility in the modeling phase,

where we can assess the impact of including versus excluding/ replacing flagged data on model performance.

3.2 Imputation of Missing Values

As can be inferred from Table 1, *Weight (kg)*, *Run (km)* and *Ride (km)* had 77, 115 and 75 missing entries respectively. Furthermore, *Sleep score*, *Resting heart rate*, *Body Battery*, *Breaths per minute (during sleep)*, *HRV-status*, *Deep sleep (mins)*, *REM sleep (mins)*, *Restless moments*, and *Duration (mins)* each had 4 missing entries (same rows). To handle missing data appropriately, we aligned the imputation strategy with the nature of each variable.

Weight Missing values for *Weight* were imputed using the average of the most recent previous (forward fill) and the next available (backward fill) values. This method was chosen based on the assumption that weight does not fluctuate drastically from day to day, and that the most reliable estimate for a missing entry lies between its adjacent values.

Run and Ride Missing values for *Run* and *Ride* were imputed with zeros. This reflects the assumption that a missing entry in these variables indicates that the activity did not occur on that day, rather than a recording error. As such, the absence of a recorded value is treated as a valid observation of zero distance.

Sleep and Heart Data Three days of data are missing across several physiological variables. These gaps appear to occur on random, isolated days and show no systematic pattern. Given the interdependence of physiological data, missing values were imputed using Multiple Imputation by Chained Equations (MICE). This method estimates missing values based on patterns in the existing data, allowing us to leverage available information and thereby avoid oversimplifying the structure underlying these variables.

4 Feature Engineering

The four time-of-day variables (e.g., *Sunset-time*) were transformed into numerical features that capture the cyclical nature of the 24-hour day. First, each time-of-day variable was converted into minutes, creating variables such as *BedTimeMinutes* and *SunsetMinutes* which represent how many minutes since midnight the event occurred. For example:

$$10{:}30\ \text{PM} = 22 \times 60 + 30 = 1350\ \text{minutes}$$

A challenge, however, is that time is cyclical and the minutes since midnight does not capture this continuity (e.g., 11:59 PM = 1439 and 12:01 AM = 1).

Therefore, we transformed the minute values using sine and cosine to reflect this cyclical relationship:

$$\text{sin\_time} = \sin\left(\frac{2\pi \cdot \text{minutes}}{1440}\right), \quad \text{cos\_time} = \cos\left(\frac{2\pi \cdot \text{minutes}}{1440}\right)$$

This maps time onto the unit circle, so that times before and after midnight are treated as close together. For instance:

$$\text{11:59 PM:} \quad \sin\left(\frac{2\pi \cdot 1439}{1440}\right) \approx -0.004, \quad \cos\left(\frac{2\pi \cdot 1439}{1440}\right) \approx 0.999$$

$$\text{12:01 AM:} \quad \sin\left(\frac{2\pi \cdot 1}{1440}\right) \approx 0.004, \quad \cos\left(\frac{2\pi \cdot 1}{1440}\right) \approx 0.999$$

This transformation allows the model to recognize time continuity and patterns across the midnight boundary.

We then began exploring sleep related features, first creating a variable *MinutesAfterSunset*, which represents how many minutes after sunset the user went to bed that day. This was done by subtracting *SunsetMinutes* from *BedTimeMinutes*, making sure to add 1440 to any *BedTimeMinutes* occurring after midnight to ensure continuity. It was also confirmed that no bed times occurred before sunset.

The second sleep related feature created was *Sleep_Fragmentation* which aimed to quantify how disrupted the user's sleep was. It was calculated by dividing the number of restless moments during sleep by the total duration of sleep. A higher value indicates more frequent disturbances relative to sleep length, giving an indication of sleep quality.

The last sleep related feature engineered was *BedtimeCategory*. This is a categorical feature that classifies each bedtime as Early, On Time, or Late. The categorization is based on how far the bedtime deviates from the average bedtime. To bound what is considered late or early, we used 0.5 of a standard deviation from the mean, which was very close to 1 hour. Thus, an early bedtime was a bedtime occurring at least one hour earlier than the average, and a late bedtime was one that occurs at least one hour later than the average bedtime. The percent split per category ended up being 45% for on time, 28% for early and 27% for late. We then used one hot encoding to convert the categorizations into separate binary columns (*BedtimeCategory_Early*, *BedtimeCategory_On Time*, and *BedtimeCategory_Late*) so that they are better used as input features in the model. After the creation of these sleep related variables, the following were dropped (*'Bed-time', 'Wakeup-time', 'BedTimeMinutes', 'WakeTimeMinutes', 'Sunrise-time', 'Sunset-time', 'SunriseMinutes', 'SunsetMinutes'*)

Other categorical features that were engineered included *DayOfWeek* and *IsWeekend*, which indicate the day of the week (coded as 0-6 for Monday to Sunday) and a binary variable for whether the day was a weekend day or not.

We also explored a feature based on physiological measurements, resulting in *Heat_Stress_Effort*, which aimed to capture the potential strain from exercising in high temperatures. It was computed by multiplying the minutes of intense exercise by the maximum daily temperature. The idea was that physical effort in hotter conditions places more stress on the body, which may impact sleep or recovery. *HR Recovery Score* was created as follows:

$$HR\ Recovery\ Score = 100 - Resting\ Heart\ Rate$$

Hence, a lower resting heart rate yields a higher recovery score. Due to parasympathetic nervous system activation during deep sleep the resting heart rate is significantly lower than during REM-sleep or wakefulness [3] [4].

Weather related features were then also explored and created. *Temperature Comfort Index* was defined as follows:

$$\text{Temp\_Comfort\_Index} = 100 - 5 \times \left|\overline{T} - 20^{\circ}\text{C}\right|$$

where $\overline{T}$ is the mean of all nightly temperature data. A perfect score of 100 represents 20 degrees Celsius and each degree deviation reduces the score by 5 points. Optimal temperatures for sleep are between 18-22 Degrees Celsius [6]. *Humidity Comfort Index* was defined as follows:

$$\text{Humidity\_Comfort\_Index} = 100 - 2 \times \left|\overline{H} - 50\%\right|$$

here $\overline{H}$ is the mean humidity across all nightly humidity data. A score of 100 indicates an optimal 50% humidity, each percentage-point deviation subtracts 2 points. Based on scientific literature on how humidity affects sleep an optimal humidity level lies at 50%. If the humidity is below 40% or above 60% increased wake-after-sleep-onset increases [8]. *Weather Stability* was created to infer how stable the weather on that day was. Increased weather instability might lead to poorer sleep quality and increased awakenings [5]. It was defined as follows:

$$\text{Weather\_Stability} = 100 - 10 \times \text{SD}(T)$$

$SD(T)$ is the standard deviation across temperature data for that night. A 1 degree Celsius reduces the stability score by 10 points. Hence, increased variability in temperature scores lead to lower stability scores. The weather features have to be treated with caution since the weather data is measured on the outside. Hence, it cannot be verified whether the same levels of temperature and humidity were present where the individual was sleeping.

Lastly, for our classical machine learning models, we computed rolling means for every numerical feature. We then created three versions of the dataset, augmented with rolling averages over windows of 3, 5, or 7 - day rolling means alongside the original columns. These three datasets will be compared in the subsequent modeling phase.

5 Classical Modeling

5.1 Random Forest

Different sliding window sizes were tested using 3, 5 and 7 as window size. The best performance was achieved with a window size of 3. The engineered dataset was chronologically ordered and split into training (70%), validation (15%) and test (15%) sets to keep temporal dependence intact and prevent data leakage. Standardization based on the training set mean and variance was applied to all predictor variables and the same transformations were applied to the validation and test set. To reduce overfitting Recursive Feature Elimination with Cross-Validation (RFECV) was applied. A Random Forest regressor (50 trees, max. depth = 3) was used as a base estimator, to approximate feature importance. RFECV iteratively removed the least important feature at each, step optimizing for MAE on the validation folds. A lower bound of 5 features was enforced to mitigate over-pruning. The features selected can be found in Figure 8a. Hyperparameter optimization was done using Optuna for 500 parallel trials [1]. The objective was minimizing Mean Absolute Error (MAE). The final hyperparameters can be found in Table 3. Lastly, the final model using the selected features and optimized parameters was then trained on the training set and tested on the validation and test set. Performance was quantified on all datasets using MAE.

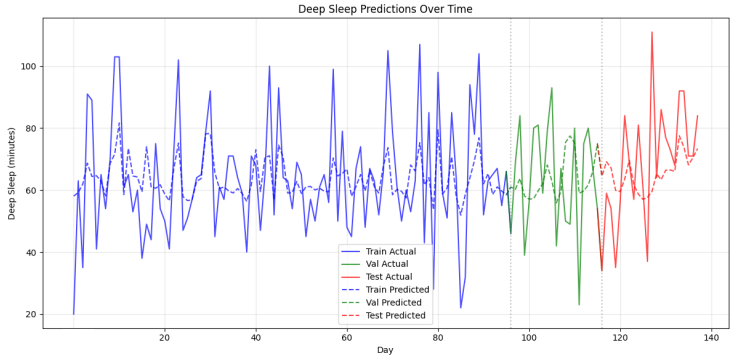


Fig. 4: Actual vs. predicted deep sleep for final random forest model

The results show a decent performance in generalizing from the training to the test set, yielding an MAE of 10.687 and 13.570, respectively. The random forest was able to make deep sleep duration predictions within a roughly 14 minute time window, compared to the observed standard deviation of roughly 18 minutes. *Wind Speed Average* was the most influential feature for prediction. At first glance it might appear counterintuitive but upon further inspection

wind speed correlates with humidity and temperature, known to influence sleep quality and therefore, deep sleep duration [10].

Hyperparameter	Value
bootstrap	True
ccp_alpha	0.018
max_depth	3
max_features	log2
max_samples	0.76
min_samples_leaf	3
min_samples_split	9
n_estimators	113
random_state	42

Table 3: Selected hyperparameters for the final random forest model.

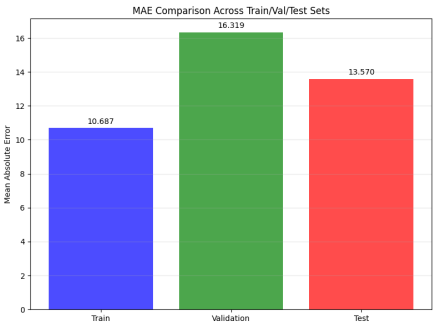


Fig. 5: Mean absolute error over datasets for the final random forest model

5.2 XGBoost

XGBoost is an optimized, regularized gradient-boosting library that builds ensembles of decision trees by sequentially fitting residuals. We kept the train - validation - test split the same as before (70/15/15) and found that the 3-day rolling-average dataset ultimately produced the best model performance. Feature selection was carried out via RFECV, iteratively removing the least important feature at each step and optimizing for MAE on time-series validation folds, with a lower bound of 8 features (chosen after manual testing of different bounds). This process yielded 80 features for the final model. Next, hyperparameters were tuned using Optuna over 500 parallel trials, minimizing MAE. The selected hyperparameters for the final model are listed in Table 4. The final XGBoost model achieved a training MAE of 13.32 min, a validation MAE of 16.30 min, and a test MAE of 14.88 min (Figure 6), indicating only a modest increase in error out of sample. The most important feature for this model was *HR_Recovery_Score*. This might be due to parasympathetic dominance lowering resting heart rate in deep sleep, making the score proxy for that stage. Predictions versus actual deep-sleep values are shown in Figure 7.

Hyperparameter	Value
n_estimators	79
max_depth	4
learning_rate	0.0021968
subsample	0.6340411
colsample_bytree	0.5973969
reg_alpha	0.0757355
reg_lambda	0.0099931
gamma	4.6358185
min_child_weight	1

Table 4: Selected hyperparameters for the final XGBoost model.

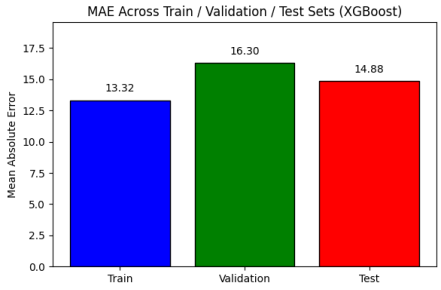


Fig. 6: Mean absolute error over datasets for the final XGBoost model model

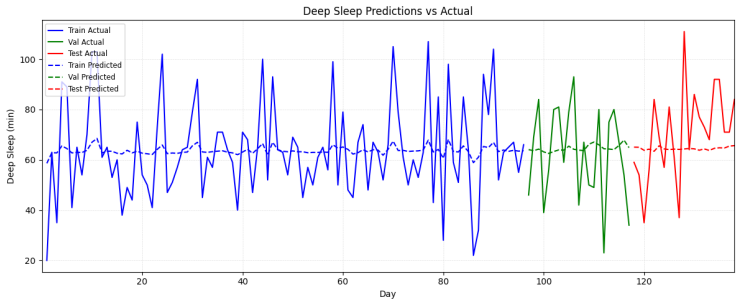
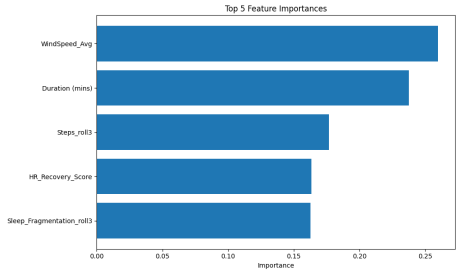


Fig. 7: Actual vs. predicted deep sleep for final XGBoost model

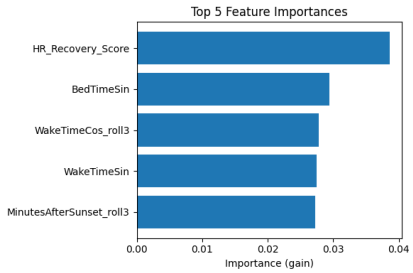
Figures 8a and 8b display the top five feature importances for the final classical machine learning models. It is evident that the engineered variable *HR_Recovery_Score* plays a critical role in both models. In addition, several three-day rolling average features also make substantial contributions to predictive performance across both models.

6 Deep Learning Modeling

For the deep learning modeling approach, we began by testing for stationarity in the time series to determine if there were trends or periodic variations in the data. Figure 9 shows deep sleep over time, with Figure 10 showing the lagged autocorrelation up to 30 lags. As can be seen, there were no significant autocorrelations beyond lag 0, confirming approximate stationarity of the data and giving us confidence to apply temporal modeling techniques. The final model chosen was the Temporal Convolutional Network (TCN) as it has been shown to have superior performance compared to recurrent models (such as RNNs) for sequence modeling [2].



(a) Random Forest



(b) XGBoost

Fig. 8: Feature importance plots for the final models. (a) Random Forest (b) XGBoost

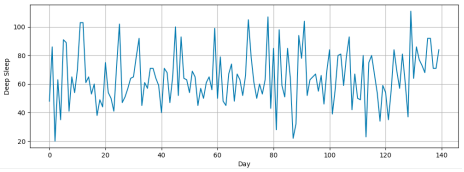


Fig. 9: Deep sleep over time

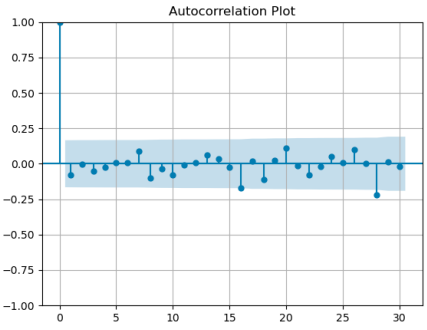


Fig. 10: Autocorrelation lag up to 30

To remain consistent with the classical approaches, the same 70/15/15 train, validation, test split was used. Furthermore, all features were standardised using scikit-learn's StandardScaler, and a fixed random seed (42) was used to ensure reproducibility. A baseline model was first trained using default hyperparameter values on the entire engineered dataset (excluding the sliding averages). Feature importance was assessed using both permutation and saliency-based methods. Permutation importance randomly shuffles each features values and measures the change in model error, whereas the saliency method computes the input gradients to identify which features are most influential for predictions. As the permutation method disrupts temporal dependencies inherent in our single-user time series, it was used only as a supplementation to the saliency method for feature selection. After running both feature importance methods, we removed features that had low importance for both, in order to reduce model complexity. Those removed were *Pressure_Max*, *MaxTemp_C*, *Pressure_Avg*, *SunriseCos*, *Pressure_Min*, *SunsetSin*, *SunriseSin*, *Breathing_Sleep_Quality*.

Hyperparameter tuning was then conducted via a grid search. Table 5 shows the hyperparameters that were tuned with the best performing values in bold.

Figure 11 shows the MAE from each set after final model training, revealing potential overfitting on the training set. Interestingly, the MAE for the test set (11.9) ended up being substantially smaller than that of the validation set (19.9). After visual inspection of the output, we noticed an outlier in the validation set on day 113 (Deep Sleep (mins) = 23) and ran a test to compare the standard deviations of deep sleep values used for predictions across the three sets. The results revealed standard deviations of 18.39, 20.97, and 18.27 for the training, validation and test sets respectively. Thus, it could be that this higher variability in the validation set is indicative of more noise relative to the training and test sets, making it harder for the model to make accurate predictions.

Hyperparameter	**Values**
Seq_length	7, **10**, 12
Num_channels_list	[16, 16], **[32, 32]**
Kernel_size	**3**, 5, 7
Dropout	0.1, **0.3**, 0.5
Learning_rate	**0.001**, 0.0005
Batch_size	**8**, 16, 32
Epochs	200, **350**, 500

Table 5: Hyperparameter grid search for TCN

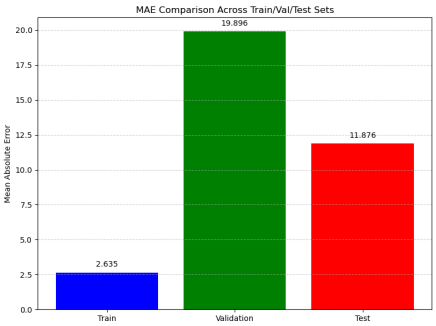


Fig. 11: MAE comparisons for the final TCN model

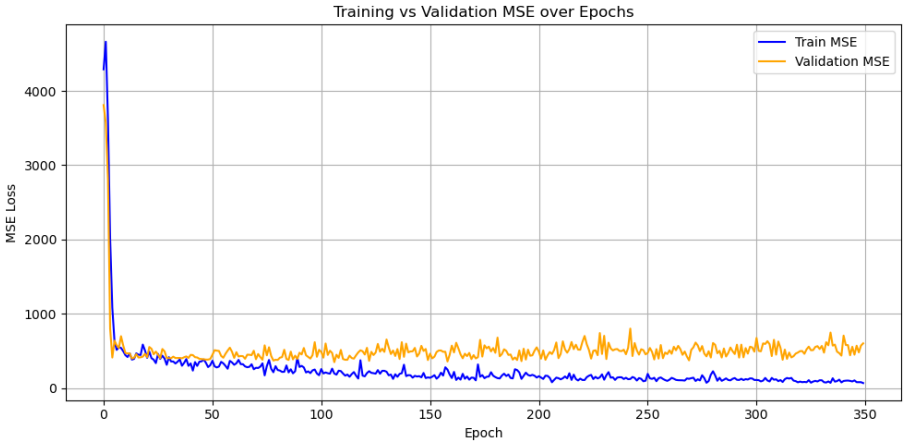


Fig. 12: Training and Validation Loss

To explore potential overfitting, we plotted a learning curve as seen in Figure 12, which show the training and validation loss over 350 epochs. While the training loss continued to decrease, the validation loss plateaued indicating a limit to the generalization of the model. Although dropout and other regularization methods were tried, the model could not perform better on the validation set than with those hyperparameter values found in Table 5.

Furthermore, Figure 13 reveals the actual vs. predicted minutes of deep sleep across training, validation, and test sets. The gaps in the graph are due to the nature the sequence-based modeling approach, where predictions can only be made after 10 (the sequence length amount) days for each set. As expected from the MAE outputs, the model does extremely well on the training set, with a drop in performance notable for the validation and test sets.

Overall, however, the model performed quite well on the test set, outperforming both classical machine learning methods which had MAEs of 13.57 (Random Forest) and 14.88 (XGboost). The TCN achieved a mean absolute error (MAE) of 11.87, meaning that its predictions deviated from the actual minutes of deep sleep by an average of just under 12 minutes. Given that the standard deviation of the test set was 18.27, the average prediction error of 11.87 corresponds to approximately 0.65 standard deviations, indicating reasonably accurate performance relative to the variability in the data.

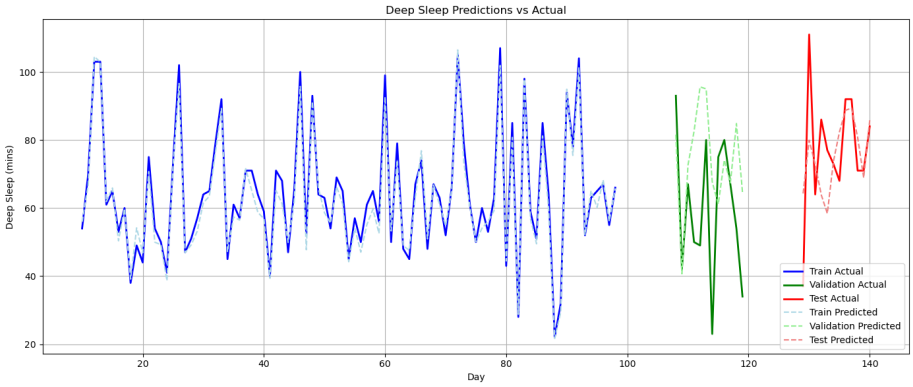


Fig. 13: Actual vs. predicted deep sleep for the final TCN model

To explore which features were important for prediction, we re-ran the gradient saliency method on the final dataset and model. Figure 14 shows the top 20 influential features. The most influential features for predicting deep sleep were *Breaths per minute* (0.6303), *IsWeekend* (0.5874) and *BedTimeCos* (0.4934). This suggests a strong link between respiratory patterns during sleep, as well as the importance of routine and weekly structure. Other top physiological contributors included *Steps*, *Max HR*, *Run (kms)* and *Minutes intense exercise*, indicating the bodies need for recovery (deep sleep) after increased physical exertion

during the day. Interestingly, weather-related variables such as *Humidity_Min* and *WindSpeed_Min* were also found in the top 20 influential features, suggesting a role of the environment for favorable sleeping conditions as outlined in earlier sections.

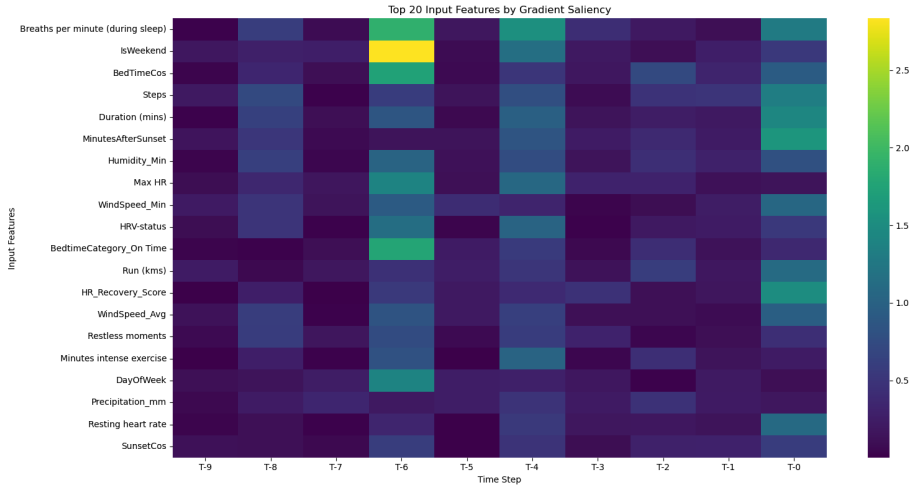


Fig. 14: Saliency map of model features

7 Conclusions

Classical machine learning models achieved reasonable mean absolute error and generalization to unseen data, but a close inspection of the actual versus predicted *DeepSleep* values reveals that they predict the mean deep-sleep duration and fail to capture day-to-day variability, although the random forest followed *Deep Sleep* trends slightly more than XGBoost. Increasing model complexity, for XGBoost and random forest, led to severe overfitting and worse performance on validation and test sets, suggesting that these algorithms do not adequately model the temporal dynamics of our data. By contrast, TNC not only reduced MAE but also produced more nuanced predictions that reflect sleep variability. Further improvements may require a larger dataset. Additionally, feature engineering was vital: many of the top predictors (e.g. *IsWeekend*, *HR_Recovery_Score*) were derived rather than raw inputs. Further, weather variables play a role in the prediction of *Deep Sleep* as suggested by scientific literature. Lastly, a critical consideration is that all data in this project stem from a single user. While we tested model performance on unseen days for that individual, we did not evaluate how well the models generalize to other people. This would be an important next step in the evaluation of our models.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). pp. 2623–2631. ACM, Anchorage, AK, USA (Aug 2019). https://doi.org/10.1145/3292500.3330701
2. Bai, S., Kolter, J.Z., Koltun, V.: An empirical evaluation of generic convolutional and recurrent networks for sequence modeling (2018), https://arxiv.org/abs/1803.01271
3. Bonnet, M., Arand, D.: Heart rate variability: sleep stage, time of night, and arousal influences. Electroencephalography and clinical neurophysiology **102**(5), 390–396 (1997)
4. Boudreau, P., Yeh, W.H., Dumont, G.A., Boivin, D.B.: Circadian variation of heart rate variability across sleep stages. Sleep **36**(12), 1919–1928 (2013)
5. Chevance, G., Minor, K., Vielma, C., Campi, E., O'Callaghan-Gordo, C., Basagaña, X., Ballester, J., Bernard, P.: A systematic review of ambient heat and sleep in a warming climate. Sleep Medicine Reviews **75**, 101915 (jun 2024). https://doi.org/10.1016/j.smrv.2024.101915
6. Harding, E.C., Franks, N.P., Wisden, W.: The temperature dependence of sleep. Frontiers in Neuroscience **13**, 336 (apr 2019). https://doi.org/10.3389/fnins.2019.00336
7. Li, C., Zhang, L., Wang, K.: Outlier detection algorithm based on gaussian mixture model. International Journal of Advanced Computer Science and Applications (IJACSA) **10**(12), 136–142 (2019). https://doi.org/10.14569/IJACSA.2019.0101218, https://www.researchgate.net/publication/338363343_Outlier_Detection_Algorithm_Based_on_Gaussian_Mixture_Model
8. Manzar, M.D., Sethi, M., Hussain, M.E.: Humidity and sleep: A review on thermal aspect. Biological Rhythm Research **43**(4), 439–457 (sep 2011). https://doi.org/10.1080/09291016.2011.597621
9. Marcin, A.: What is deep sleep and why is it important? (Mar 2023), https://www.healthline.com/health/deep-sleep, medically reviewed by Daniel Murrell, M.D.; accessed June 20, 2025
10. Okamoto-Mizuno, K., Mizuno, K.: Effects of thermal environment on sleep and circadian rhythm. Journal of physiological anthropology **31**, 1–9 (2012)
11. The Weather Company: Monthly weather history for amsterdam, netherlands. https://www.wunderground.com/history/monthly/nl/amsterdam (2025), accessed: 2025-06-05
12. Time and Date AS: Sunrise and sunset times for amsterdam, netherlands – february 2025. https://www.timeanddate.com/sun/netherlands/amsterdam?month=2 (2025), accessed: 2025-06-05
13. Weather and Climate: February 2025 daily weather monitor data (station 06240) (Feb 2025), https://www.weatherandclimate.info/monitor/?id=06240&month=2&year=2025, accessed: 2025-06-05